CONDITIONING OF
DIFFERENTIAL ALGEBRAIC EQUATIONS
AND
NUMERICAL SOLUTION OF
MULTIBODY DYNAMICS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de Rector Magnificus, prof.dr. J.H. van Lint, voor
een commissie aangewezen door het College
van Dekanen in het openbaar te verdedigen op
donderdag 18 april 1996 om 14.00 uur

door
PATRICK MARTIN ELIZABETH JOZEF WIJCKMANS
Geboren te Kerkrade
Dit proefschrift is goedgekeurd door de promotoren:

prof. dr. R.M.M. Mattheij
en
prof. dr.ir. J.S.H.M. Wismans

CIP-DATA KONINKLIJE BIBLIOTHEEK, DEN HAAG

Wijckmans, Patrick Martin Elizabeth Jozef

Conditioning of differential algebraic equations
and numerical solution of multibody dynamics /
Patrick Martin Elizabeth Jozef Wijckmans. - [S.l. : s.n.]
Thesis Technische Universiteit Eindhoven. - With index, ref.
ISBN 90-386-0457-2
Subject headings: differential algebraic equations /
numerical mathematics, computational dynamics.
CONTENTS

Preface ix

1 Scope of the Thesis 1
   1.1 Introduction 1
   1.2 Multibody Dynamics and Formalisms 3
       1.2.1 History of Multibody Dynamics 3
       1.2.2 The Choice of a Set of Variables 4
       1.2.3 The Choice of the Formulation of the Dynamics 4
       1.2.4 Open Loop and Closed Loop Systems 6
       1.2.5 Other Practical Aspects 6
       1.2.6 Multibody Programs 7
   1.3 Occurrence of Differential Algebraic Equations 7
       1.3.1 Historical Overview 7
       1.3.2 Characteristics of DAEs 8
   1.4 Objectives 9
   1.5 Contents of this Thesis 11

2 Multibody Systems 13
   2.1 Basic Concepts for Generation of Equations of Motion 13
       2.1.1 Primitive Equations of Motion 13
       2.1.2 Incorporation of Constraints 14
       2.1.3 Augmentation Method 16
       2.1.4 Elimination Method 18
   2.2 Recursive Formulation for Constrained Mechanical Systems 19
       2.2.1 Topology 20
       2.2.2 Kinematic Relations 21
       2.2.3 Open Loop Systems 24
2.2.4 Closed Loop Systems ........................................ 26
2.2.5 Recursive Algorithm ....................................... 29
2.3 Conclusions .................................................... 30

3 Differential Algebraic Equations ................................ 31
3.1 Theory of Differential Algebraic Equations ................. 31
  3.1.1 Introduction ............................................... 31
  3.1.2 Linear DAEs with Constant Coefficients ............... 32
  3.1.3 Nonlinear Systems ....................................... 35
  3.1.4 Semi-Explicit Systems ................................... 37
  3.1.5 Applications .............................................. 38
3.2 Multistep Methods ............................................ 40
  3.2.1 Constant Coefficient DAEs .............................. 41
  3.2.2 Index One Systems ....................................... 43
  3.2.3 Semi-Explicit Index Two Systems ....................... 44
  3.2.4 Index Three Systems of Hessenberg Form .............. 45
3.3 Runge-Kutta Methods ......................................... 46

4 Conditioning of Differential Algebraic Equations of Index One .... 49
  4.1 General Linear DAEs of Index One ......................... 49
  4.2 Semi-Explicit Index One DAEs .............................. 52
  4.3 Conditioning for Almost Singular D ....................... 53
  4.4 Asymptotic Analysis of Almost Higher Index DAEs ....... 61
  4.5 Index One DAEs Close to Higher Index DAEs .............. 64

5 Conditioning of Differential Algebraic Equations of Index Two .......... 69
  5.1 Introduction .................................................. 69
  5.2 Semi-Explicit Index Two DAEs .............................. 70
  5.3 Conditioning For Almost Singular Matrix CB ............ 73
  5.4 Perturbations of the Coefficients ......................... 76

6 Regularization and Stabilization ................................ 85
  6.1 Solution Techniques For Higher Index DAEs ............... 86
    6.1.1 Regularization Methods ............................... 87
    6.1.2 Generalized Coordinate Partitioning ................. 89
    6.1.3 Projection Methods ................................... 90
    6.1.4 Overdetermined Differential Algebraic Equations .... 91
  6.2 The Iteration Matrix ....................................... 92
    6.2.1 Conditioning of the Iteration Matrix ............... 92
    6.2.2 Effect of Rounding Errors on Solution Components .... 96
    6.2.3 Stabilized Index Reduction ......................... 99
  6.3 Error Estimation for Step-size Control and Termination of Newton Iterations ........... 102
  6.4 Application To Multibody Systems ......................... 105
CONTENTS

7 Discontinuities in Mechanical Systems 109
  7.1 Numerical Problems due to Shocks and Discontinuities .... 109
  7.2 Treatment of Discontinuities .......................... 111
  7.3 Occurrence of Discontinuities in Multibody Dynamics ...... 113
  7.4 Usage of The Switching Functions ...................... 115
     7.4.1 Root Finding .................................. 116
     7.4.2 An Algorithm Based on Monitoring The Switching Functions ... 118
  7.5 Use of the Method in Mechanical Problems ............... 119

8 Conclusions and Discussion 127
  8.1 Conclusions ........................................... 127
  8.2 Discussion ........................................... 129

A Reduction of Equations of Motion for a Chain 131

B Reduction of Equations of Motion for a Closed Loop System 133

C Relation between the fundamental solution and the Green function of an
   ODE and a slightly perturbed one .......................... 137

References 139

Index 147

Samenvatting 151

Curriculum Vitae 155
This thesis is the result of the research I have done during the past years at the Scientific Computing group of the Eindhoven University of Technology.

At this place, I wish to express my gratitude to all people who have contributed to my work in one way or another. I thank prof.dr. Mattheij for his stimulating support and the many fruitful discussions throughout all these years. His enthusiastic encouragement helped me a lot in difficult times. I am also grateful to dr.ir. Koppens for our discussions about multibody systems and for providing some insights into MADYMO. I would like to thank prof.dr.ir. Wismans for his critical remarks about the presentation of my work. In addition, I thank the Road Vehicles Institute of TNO for partially financing this research. Naturally, I would like to thank all my friends and colleagues for their support and collaboration and for providing a nice atmosphere to work. At last, I would like to thank Myriam for her everlasting care and support.

Patrick Wijckmans

Eindhoven, February 1996.
1

SCOPE OF THE THESIS

1.1 INTRODUCTION

Multibody systems are mechanical systems, consisting of a finite number of bodies, both rigid and elastic, which are interconnected in a way that allows for a large relative motion between the bodies. These interconnections consist of force elements, such as springs and dampers, and joints. Joints constrain the relative motion between the interconnected bodies and as a result they are the cause of constraining forces. A wide variety of mechanical systems can be modelled in this way, such as motor vehicles (cf. Figure 1.1), robots, spacecraft, antennas, and the human body.

![Figure 1.1 A vehicle.](image)

Multibody analyses have been applied extensively in bio-dynamic modelling. The desire to have a better understanding of the dynamic behaviour of muscle-skeletal sys-
tems has led to many of the major developments in multibody systems theory. Multibody models of man (cf. Figure 1.2) have been used in several areas of bio-mechanics like the development of safety devices, such as seat belts and air bags, the design of prosthetic limbs, and bio-mechanics for sports. Another important application of multibody sys-

![Figure 1.2](image)

**Figure 1.2** A frontal impact. This figure depicts a crash safety simulation, which displays the response of a dummy in a frontal crash, where the dummy is restrained by a passenger air bag and a three-point belt.

tems theory is the analysis of robots and mechanisms. Such systems are composed of connected bodies and are ideally suited for modelling as multibody systems. Figure 1.3 depicts a typical robot. Simulation of the motion of a multibody system is useful for various problems of *dynamic analysis*. Of interest is particularly the motion of the multibody system, i.e. the positions, the velocities and the accelerations, but also the internal forces under the influence of the externally applied forces. Such dynamic simulations are an important part of computer aided design. They give the designer or engineer a powerful
1.2. Multibody Dynamics and Formalisms

1.2.1 History of Multibody Dynamics

The first person to formulate the equations to describe multibody systems for the human body was Fischer (1906) (cf. [26]), who modelled the human body as a system of three
coupled rigid bodies. However, he was unable to solve the resulting equations. Generally, the motion of multibody systems is described by differential equations, often coupled with algebraic constraint equations that describe the interconnection between adjacent bodies. These equations must be satisfied by the relative motion of the interacting bodies and the Lagrange multiplier associated with the aforementioned constraints. In classical mechanics these equations are reduced to a system of ordinary differential equations (ODEs), the so-called state space representation or Lagrange equations of type two. In general, large displacements are possible and the descriptive equations are highly nonlinear. The reduction to a state space representation may require strong simplifications of the mechanical models. As a result, we can only formulate the governing equations by hand for very simple mechanical systems and there are only a few mechanical systems that can be completely solved analytically. This gave rise to the need for reliable and efficient numerical methods for the simulation of mechanical systems. That is why in the early sixties the first multibody formalisms were developed, i.e. methods for the generation of the descriptive equations. There were two separate impulses for this development. The first was the increasing power offered by digital computers, and the other was the need for detailed analysis of mechanical systems in the design of spacecraft and high-speed mechanisms. The study of specific cases was started then and several special purpose programs were written. Towards 1965 increasing attention was paid to the construction of general purpose multibody computer programs that are capable of simulating very broad classes of multibody systems. Since then several of these multibody programs have been developed.

1.2.2 The Choice of a Set of Variables

While developing such multibody programs there are some very important issues that have to be considered. First, in the modelling process the choice of the set of variables used to represent the motion of the system is crucial. This set of variables should define the positions, the velocities and the accelerations of the system in a unique way at each time. The most important types of coordinates are absolute coordinates, i.e. Cartesian coordinates, and relative coordinates or state variables. The absolute coordinates define the position and the velocity of a body with respect to an inertial reference frame. This results in a maximal set of Cartesian coordinates for each body. The relative variables represent the position and the velocity of a body in terms of relative motions between the interconnected bodies and result in a minimal set of variables. The choice of a set of variables is closely related to the formulation of the dynamics of the mechanical system, which will be described in the next subsection.

1.2.3 The Choice of the Formulation of the Dynamics

Another important aspect is the choice of the formulation of the dynamics. The dynamic equations are based on the laws of Newton and Euler. For the development of multibody formalisms various methods for the generation of the equations of motion are used, i.e. Newton-Euler equations, Lagrange’s equations of the first and second kind.
and d'Alembert's principle of virtual work. All these formalisms lead to different, but
equivalent, formulations of the equations of motion. The relative advantages or disad-
vantages of the approaches depend upon the particular dynamic method being used, and
the method used to organize the complex geometry. In general, one can determine two
approaches, viz. the *augmentation method* and the *elimination method*, depending on
whether one wants to augment the constraint equations to the dynamic equations lead-
ing to the so-called *descriptor form* or whether one wants to eliminate the constraints
from these equations, resulting in a state space representation. In the first approach one
chooses a set of absolute coordinates. As a result, this approach will lead to a maxi-
mal number of descriptive equations, i.e. dynamic as well as constraint equations. How-
ever, the generation of the resulting equations of motion will be rather simple in this
approach. As an example, consider a double pendulum (cf. Figure 1.4) which is kept
as a reference model during the whole thesis. This double pendulum is a compound

![Figure 1.4 The double pendulum](image)

of two coupled uniform rigid rods, denoted as bodies \( B_1 \) and \( B_2 \), respectively, moving
under the influence of gravity. The centres of mass of \( B_1 \) and \( B_2 \) are \( M_1 \) and \( M_2 \), re-
spectively. For this system the position of \( M_1 \) is given by the absolute Cartesian co-
ordinates \( x_1 \) and \( y_1 \) and the orientation of rod \( B_1 \) is given by \( \phi_1 \), and likewise for \( B_2 \) by
\( x_2 \), \( y_2 \) and \( \phi_2 \). However, this system has only two degrees of freedom, viz. the rotation
angles \( \phi_1 \) and \( \phi_2 \), because the system is constrained by the requirement that rod \( B_1 \) has
its pivot pin in \( O \) and that the pivot pin of rod \( B_2 \) coincides with the pivot on the distal
end of rod \( B_1 \). This implies that the dynamic equations have to be augmented with con-
straint equations, while unknown Lagrange multipliers account for the constraint loads.
Hence, the resulting equations of motion (which will be derived more in detail in Chap-
ter 2) form a system of differential equations (i.e. the dynamic equations) together with
algebraic equations (i.e. the constraint equations) and is therefore called a system of *dif-
ferential algebraic equations* (DAEs).

Alternatively, for the double pendulum shown in Figure 1.4 the rotation angles \( \phi_1 \)
and \( \phi_2 \) can be chosen as relative coordinates, resulting in a system of ODEs (this is elab-
orated in Chapter 2). The second method uses relative variables. This minimal set of
*joint coordinates* leads to fewer equations with higher complexity, since it is possible to
eliminate constraint equations and constraint loads.
1.2.4 Open Loop and Closed Loop Systems

Relative variables are especially effective for open loop systems where the bodies form tree configurations. These open loop systems appear naturally in the description of spacecraft and robots (cf. Figure 1.3). The compound double pendulum of Figure 1.4 is an open loop system. The resulting equations of motion form a system of nonlinear (ODEs).

Many mechanical systems, however, are closed loop systems containing closed chains, implying that some elements of the multibody system are connected in more than one way. The treatment of this class of systems, however, is far more complicated than the treatment of open loop systems. Systems with closed chains can be transformed into tree configuration by cutting selected joints (an idea developed by Wittenburg [84]), the so-called cut joints between bodies. This implies that the relative coordinates are not independent anymore, since they are subject to these additional cut joints. Consider a crank-slider mechanism as shown in Figure 1.5. This mechanism can be brought into tree configuration by cutting the joint constraining the distal end of \( B_2 \) to move along the \( x \)-axis. Here, the rotation angles \( \phi_1 \) and \( \phi_2 \) are not independent. Hence, the dynamic equations have to be augmented with a constraint equation that constrains the distal end of \( B_2 \) to the \( x \)-axis, and again, Lagrange multipliers account for the constraint loads. Therefore, the resulting equations of motion form a system of DAEs.

1.2.5 Other Practical Aspects

Other important aspects in the simulation of multibody systems include the reduction of computer time required for the simulation, which is most important in real-time simulations, and the minimization of the amount of data storage, since realistic mechanical models can be huge and very complex. Therefore, there is a growing interest in multibody formalisms that have a high potential for parallel computation (cf. [6, 23]).

Another practical problem in the dynamic analysis of multibody systems is the occurrence of discontinuities. These discontinuities appear in the modelling of e.g. crashes, hysteresis, dry friction and contact problems. The appearance of discontinuities cause severe problems during the numerical solution of such problems. An important requirement for a multibody code is that it can deal with these discontinuities in a robust and efficient way.
1.3 OCCURRENCE OF DIFFERENTIAL ALGEBRAIC EQUATIONS

In the last decade there has been a growing interest in the modelling of flexible bodies in multibody systems (cf. [16, 51–53]). The model of an air bag shown in Figure 1.2 is an example of such a flexible body. In the referred works flexibility is restricted to small deformations of bodies experiencing large displacements.

1.2.6 Multibody Programs

As a result of these observations, several general purpose multibody programs have been developed. These kinds of programs generate both the kinematic and dynamic equations, solely based on input data describing the way the bodies are interconnected, the mechanical and geometrical properties of the bodies and the interactions between them, together with the system state at the outset. Afterwards they integrate the resulting equations of motion. A variety of powerful new algorithms that efficiently generate the highly nonlinear equations of motion of multibody systems has been developed. This made it possible to derive the governing equations for very complex and realistic mechanical systems, even, so that very detailed models of machines and robots for example can be simulated.

Various methods are used in these multibody programs. Cartesian coordinate formulations are the basis for codes like ADAMS (cf. [15]) and DADS (cf. [80]), which are widely used in industry. They express the equations of motion in descriptor form and they result in a large set of highly sparse equations. Roberson (cf. [72]) and Wittenburg (cf. [84]) introduced graph theoretical methods, with cut joint concepts that lead to spanning trees and a minimal set of generalized coordinates and constraint equations. These methods form the basis for all sorts of new computational formalisations. They have led to the generation of recursive formalisms (cf. [1, 5, 6, 49, 73, 79]) that build upon the topological relative coordinate foundation. These recursive formalisms generate the equations of motion in a very efficient way. The general multibody program MADYMO (cf. [58]) uses such a recursive formulation. It has several features for crash analyses. MADYMO can be used to simulate the behaviour of crash-victims. Since it contains a finite element module, one can also study the interaction between the victims and the deformable vehicle structures as well as the safety devices. This combination of a multibody package and a finite element method is especially important when the interaction with highly deformable structures, like air bags, or vehicle interior padding is simulated (cf. Figure 1.2).

1.3 OCCURRENCE OF DIFFERENTIAL ALGEBRAIC EQUATIONS

1.3.1 Historical Overview

Section 1.2 shows that, depending upon the particular dynamic method and the geometric properties, the resulting equations of multibody systems are composed of DAEs or ODEs. DAEs also arise in many other application areas, including electrical networks, flows of incompressible fluids, control theory, robotics and chemical reaction kinetics.
One can think of DAEs as systems of differential equations coupled with constraining equations. The first paper on the numerical solution of DAEs was written by Gear (cf. [31]) in 1971. Particularly, the so-called backward differentiation formulae (BDF) (cf. Chapter 3) appeared to be effective for the solution of these systems. Only in the 1980's did the systematic study of numerical methods for the solution of DAEs begin and over the last few years there has been growing research activity in this area. Petzold (cf. [65]) has shown that DAEs can differ substantially from ODEs, a fact that produces great difficulties for the numerical integration of DAEs. In fact, the numerical solutions of DAEs are far more difficult than the solutions of ODEs. Some DAEs can be solved numerically by methods developed for the solution of stiff ODEs, whereas other DAE systems can not be solved by such ODE solvers. The research activities of the 1980's contributed much to the understanding of the nature of DAEs. The most important results over these years were summarized in four interesting monographs [9, 40, 41, 43]. They provide good insight into both the mathematical structure of DAEs and the analysis of numerical methods applied to DAEs.

1.3.2 Characteristics of DAEs

As stated earlier, DAEs are difficult to solve numerically. To explain this briefly, consider some simple DAE systems:

\[
\begin{align*}
    x + y &= u, \\
    x - y &= v,
\end{align*}
\]

where \( u \) and \( v \) are given forcing functions. The latter DAE has the following solution

\[
\begin{align*}
    x(t) &= \exp(\theta x(0) + \int_0^\theta \exp[s(\theta + v(s))]ds, \\
    y(t) &= x(t) - v(t).
\end{align*}
\]

One observes that the solution is of the same form as for ODEs, i.e. the solution depends on the forcing terms themselves but not on derivatives of them. Therefore, this DAE is called a DAE of index one. However, there is one difference with respect to ODEs, since the initial value \( y(0) \) cannot be chosen freely. It has to satisfy the relation \( y(0) = x(0) - v(0) \), whereas one may choose an arbitrary initial value for \( x(0) \). In fact, this DAE will not exhibit any problems, except the usual ones in numerically solving ODEs. A somewhat more complicated DAE is given by the following system

\[
\begin{align*}
    x + y &= u, \\
    x &= v,
\end{align*}
\]

for given forcing functions \( u \) and \( v \). It is obvious that this DAE has the solution

\[
\begin{align*}
    x &= v, \\
    y &= u - v.
\end{align*}
\]
1.4. Objectives

Contrary to ODEs or DAEs of index one, the solution in this case depends on the first derivative of the forcing function \( v \). Furthermore, both initial values \( x(0) \) and \( y(0) \) cannot be chosen freely. The index of this DAE is two and the system appears to be more difficult to solve numerically than the system of index one or a system of ODEs. Next, consider the DAE system

\[
\begin{align*}
\dot{x} + y &= u, \\
\dot{y} + z &= v, \\
x &= w,
\end{align*}
\]

(1.1)

for given functions \( u, v \) and \( w \). The resulting solution is

\[
\begin{align*}
x &= w, \\
y &= u - w, \\
z &= v - \dot{u} + \dot{w}.
\end{align*}
\]

For this system the solution not only depends on the first derivative of the forcing functions, but even on the second derivative of the forcing function \( w \). As a consequence, this DAE system has index three and it is even more difficult to solve numerically than the aforementioned index two system. In DAE theory the index concept gives a classification of DAEs. The index will characterize the numerical difficulty of a DAE.

Any textbook on multibody systems (e.g. [45, 73, 84]) will show that the equations of motion for systems containing only kinematic constraints generally are of the following form

\[
\begin{align*}
\dot{\mathbf{q}} &= \mathbf{v}, \\
M \ddot{\mathbf{q}} + \mathbf{J}_\lambda \dot{\lambda} &= \mathbf{g}, \\
0 &= \phi(\mathbf{q}, t),
\end{align*}
\]

(1.2)

where \( \mathbf{q} \) denotes the generalized positions, \( \mathbf{v} \) denotes the generalized velocities and \( M \) is the generalized mass matrix that is positive definite. The applied and outer forces are given by \( \mathbf{g} \). The matrix \( \mathbf{J}_\lambda := \frac{\partial \phi}{\partial \lambda} \) is the Jacobian of the constraint equations. The Lagrange multipliers \( \lambda \) account for the constraint loads. Comparing the index three DAE (1.1) in the variables \([x, y, z]^T\) to equation (1.2) in the variables \([\mathbf{q}^T, \mathbf{v}^T, \lambda]^T\), it is easy to see (by identifying \( x \) with \( q \), \( y \) with \( v \) and \( z \) with \( \lambda \)) that (1.2) is a DAE of index three. As a consequence, they are very difficult to solve numerically.

1.4 Objectives

The main reason for writing this thesis was to develop a reliable and efficient numerical method for approximating the dynamics of multibody systems with closed loops. Multibody dynamics often give rise to DAEs with forcing functions exhibiting discontinuities in the form of finite jumps, either in the function itself or in some derivatives of it. Therefore, the method should adapt to these discontinuities.
To accomplish our goals we first have to study the structure of the dynamic equations of multibody systems and more especially for closed loop systems. Since they generally constitute a system of DAEs, a thorough study of such equations is needed. An important topic in DAE theory is the \textit{conditioning} of DAEs, i.e. the sensitivity of the equations to small changes in the system. The conditioning of a DAE can be considered by deriving a state space equation for the DAE, resulting in an ODE. The conditioning of this state space ODE can be studied as in standard theory for ODEs then. However, for DAEs some important quantities have to be introduced that are not needed in ODE theory. This is due to the different nature of such equations compared to ODEs. In Section 1.3 the importance of the index concept of a DAE was already pointed out. It appears that index $\nu$ DAEs may effectively behave like DAEs of index higher than $\nu$, i.e. DAEs of index one may behave as complicated as DAEs of index two or higher for example. Unfortunately, it appears that in the case of DAEs of higher index perturbations of the equations, especially in some coefficients, may lead to large perturbations in the solution. This might cause severe problems during the numerical solution of such systems. In fact, it would be easier to solve the associated reduced higher index problem. We already pointed out that the numerical solution of DAEs of a higher index is rather difficult. In general the integration of such problems by methods for the solution of stiff ODEs is not possible, since most of them are only convergent for index one DAEs and a loss of the approximation order for the algebraic variables occurs. BDF methods, however, are shown to converge for some classes of DAEs of a higher index, such as Hessenberg forms (cf. Chapter 3) which is the general form of the equations of motion for multibody systems, although they exhibit problems due to ill-conditioning of the iteration matrix, and the order and step-size control may fail. Reducing the index to one by differentiation of the constraints yields a problem that can be integrated by such a numerical method. A major disadvantage of this approach is that the numerical solution will drift away from the constraints, since it does not satisfy the original constraints. Therefore a numerical method has to be developed, that minimizes this drift-off effect, such that higher index DAEs, generated in e.g. multibody dynamics, can be integrated numerically.

As stated in the objectives the numerical method for the solution of the equations of motion of mechanical systems should solve systems exhibiting discontinuities, since in multibody dynamics discontinuities occur very often, e.g. in the modelling of impact (cf. also Figure 1.2), hysteresis, Coulomb friction, etc. The lack of smoothness, due to the discontinuities, is the cause of numerical difficulties, since the required differentiability in the convergence analysis is not present. As a consequence, the step-size selection of a numerical method may break down and there is no safe local error estimation anymore. During the passage of the singularity, the method may become very inefficient because of repeated step-size reductions. These problems can be circumvented by the use of switching functions that determine whether the integrator passes a discontinuity. These functions can be used either to localize the singularity or to reduce the step-size of the numerical method. Then the integration is restarted at the singularity, or it is continued with reduced order and step-size. In this way the discontinuity is handled in a more efficient way.
1.5 CONTENTS OF THIS THESIS

In this section we briefly outline the contents of this thesis. In Chapter 2 the mathematical formulation describing the kinetics and dynamics of multibody systems is derived. The laws of Newton and Euler form the basis for the dynamic equations. From these basic concepts the dynamic equations for mechanical systems can be generated. The constraint equations, describing the interconnections between several elements of the multibody system, can be incorporated in two different ways. In the augmentation method the constraints are added to the dynamic equations. Application of the method of Lagrange multipliers implies that the constraining forces are represented by the unknown Lagrange multipliers. This augmentation method results in the descriptor form of the equations of motion. The resulting system is a system of DAEs. In the elimination method, one chooses a set of relative coordinates describing the motions of bodies relative to each other. Use of this set of so-called state variables eliminates both the constraint equations and the associated forces, yielding the state space formulation forming a set of ODEs. However, this method only generates a system of ODEs for open loop systems (cf. Figure 1.4). The generation of these equations can be performed in a very elegant and efficient way using a recursive formulation of the multibody system. In cases where the multibody system contains closed loops, one can also use this recursive method to generate the dynamic equation. For this type of problem one has to cut some joints in order to retrieve an open loop system. Thereby one introduces cut joints. This implies that the resulting equations of motion now form a system of DAEs, since the relative coordinates are not independent anymore because of the cut joints that have to be accounted for.

Chapter 3 focuses on the basic theory of DAEs. It explains the index concept and gives some related definitions of the index, such as the differential index and the perturbation index. The index gives a classification of DAEs and plays a key role in the study of existence and uniqueness of solutions of DAEs. In a way, the index indicates how much a DAE differs from an ODE. A characterization of DAEs regarding their index and their structure is given. In addition, the numerical solution of DAEs by direct methods, such as multistep and Runge-Kutta methods is studied. In general, only DAEs of index one can be solved directly. It appears that the higher the index, the greater the numerical difficulties.

The objective of Chapter 4 is to study the question of conditioning of DAEs of index one. After briefly considering general linear DAEs the more transparent case of semi-explicit linear index one DAEs is studied. In this manner we can analyse the influence of perturbations of the equations on the solution, which obviously is very important for the numerical solution of DAEs. We will show that index one DAEs may be close to a DAE of a higher index, i.e. the solution of such index one systems effectively behaves like the solution of a higher index DAE. Consequently, such equations generally are ill-posed and are therefore difficult to solve numerically. These kinds of systems practically behave as if they were of an index higher than two, meaning that it is important to know the effective index of the DAE.

Chapter 5 deals with the conditioning of index two DAEs and explains their solution
behaviour. We will show that index two DAEs may behave like DAEs of index three or higher. For such systems it will appear to be very interesting to consider perturbations of the coefficients.

The subject of Chapter 6 is the study of solution methods for the equations of motion for multibody dynamics. Standard methods for the numerical solution of ODEs are generally not suited for the solution of DAEs of index two or higher. Lowering the index by differentiation will not help, since index reduction gives rise to drift, which can make the numerical solution completely useless. Therefore, numerical methods have to be developed that alleviate this drift. The condition number of the iteration matrix of such methods is studied in detail, both for well-conditioned systems as well as for systems close to DAEs of higher index. Numerical methods applied to the latter type of problems can suffer from problems with respect to stability for example. Luckily however, BDF methods will exhibit these problems to a smaller extent. We will show how this sort of problems can be reduced by a particular stabilization technique. This will be demonstrated by some numerical examples. Next, applications of multibody systems are presented. Examples of ill-conditioned multibody systems will be shown and the stabilization technique of the previous chapter will be applied.

Shocks and discontinuities frequently occur in the dynamics of multibody systems and cause great difficulties for the numerical solution, i.e. solution methods for ODEs as well as for DAEs may become very inefficient or may even fail due to the presence of discontinuities. In Chapter 7 attention will be paid to the numerical simulation of such problems in order to develop methods suitable for the efficient and robust simulation of discontinuous differential equations. Applications of discontinuous mechanical systems, i.e. systems exhibiting impacts, hysteresis and Coulomb friction for example, are presented.

In Chapter 8 the achievements in this study will be reviewed in view of the objectives stated in Section 1.4. We conclude with a brief discussion and some suggestions for further research and development.
Equations of motion for multibody systems may be obtained by various formalisms. The basic approaches for generating these equations are the augmentation method and the elimination method. The augmentation method generates the descriptor form of the equations of motion and results in a system of differential algebraic equations. For open loop systems a set of ordinary differential equations is yielded by the elimination method. These methods are reviewed. Recursive methods for the generation of the equations of motion for mechanical systems significantly reduce computer time needed for generating the dynamic equations. A recursive formulation for obtaining the equations of motion, both for open loop systems and for closed loop systems, is given.

2.1 BASIC CONCEPTS FOR GENERATION OF EQUATIONS OF MOTION

Multibody simulation programs generate the multibody system equations from a description of the system elements and the system topology. These equations can be generated in various forms. Earlier work in obtaining equations of motion for mechanical systems can generally be divided into two basic approaches, viz. the augmentation method and the elimination method. In this section, methods for the generation of multibody system equations are discussed.

2.1.1 Primitive Equations of Motion

Following the concepts of classical mechanics, we assume that there is an inertial frame, such that the equations of motion, based on Newton's second axiom for translational motions and Euler's axiom of moment of momentum, hold for a system of \( n \) rigid bodies, denoted by \( B_i \) (\( i = 1, 2, \ldots, n \)). Let \( m_i \) be the mass of body \( B_i \) and \( J_i \) be the inertia matrix of \( B_i \) with respect to its centre of mass. The absolute position vector of the centroid
of \( B_i \) is denoted by \( r_i \), and the absolute angular velocity of \( B_i \) is given by \( \omega_i \). Let \( f_i \) denote the total force acting upon \( B_i \), and \( n_i \) denote the resultant moment on body \( B_i \) with respect to its centroid. With respect to the inertial frame, the equations read

\[
M_i \ddot{r}_i = g_i, \quad i = 1, 2, \ldots, n,
\]

where \( M_i := \begin{bmatrix} m_i & 0 \\ 0 & I \end{bmatrix} \) denotes the generalized mass matrix, \( \dot{r}_i \) is the global velocity of body \( B_i \), and \( g_i := \begin{bmatrix} \dot{e}_i \\ -m_i \ddot{e}_i + 2m_i \ddot{e}_i \end{bmatrix} \) is the generalized load\(^1\) vector. This can be illustrated by the following

**Example 2.1** Figure 2.1 shows a pendulum, pivoted at point \( O \). The pendulum consists of a rod of length \( 2l \) with centre of mass \( M \) and mass \( m \). Its angular moment of inertia with respect to the centre of mass is given by \( J = \frac{1}{2} ml^2 \). Under influence of gravity the pendulum will move in the \( x - y \) plane. The dynamic equations (2.1) are given by

\[
\begin{bmatrix} m \\ m \\ J \end{bmatrix} \begin{bmatrix} \dot{e} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} mg + r_f' \\ -m \ddot{e}_i + 2m \ddot{e}_i \end{bmatrix},
\]

where \( r_f' \) is the reaction force working on the rod in point \( O \).

2.1.2 Incorporation of Constraints

A multibody system is a collection of rigid bodies, arranged such that relative motion between the bodies is possible. Any set of variables that uniquely specifies the position and orientation of all bodies in the system, i.e., the configuration of the mechanism, is called a set of generalized coordinates. Since rigid bodies building a mechanism are interconnected by joints, there are equations of constraints which relate generalized coordinates. Therefore, generalized coordinates are generally dependent. Incorporation of constraints can be obtained by applying the variational equations of motion (see [94]), i.e.,

\[
\delta q^T (Mq - g) = 0,
\]

with arbitrary infinitesimal displacements \( \delta q \). Here, a composite generalized coordinate vector, a composite generalized mass matrix\(^2\), and a composite generalized load vector

---

\(^1\)In most literature in mechanics, generalized loads are denoted by the symbol \( Q \).

\(^2\)\( M \) is the generalized mass matrix with respect to the generalized coordinates \( q \). This implies that this matrix may differ from the mass matrix used in equation (2.1).
2.1. Basic Concepts for Generation of Equations of Motion

are defined by

\[ q := [q_1^T, q_2^T, \ldots, q_n^T]^T, \]

\[ M := \text{diag}(M_1, M_2, \ldots, M_n) \text{ and} \]

\[ g := [g_1^T, g_2^T, \ldots, g_n^T]^T, \]

respectively.

The loads in equation (2.2) include the unknown generalized constraint loads, which have to be exerted by the constraints in order to compel the system to fulfill the kinematic conditions. All loads other than constraint loads are called generalized applied loads. These are either explicitly known or can be formulated explicitly in terms of the generalized coordinates. Let the generalized loads \( g \) be divided into the generalized applied loads \( g^a \) and the generalized constraint loads \( g^c \), i.e., \( g = g^a + g^c \). Using the preceding notation, equation (2.2) may be expressed as

\[ \delta q^T [M\dot{q} - g^a] - \delta q^T g^c = 0, \quad (2.3) \]

for arbitrary \( \delta q \). Here, \( \delta q^T g^c \) is the so-called total virtual work of constraint loads acting on all bodies in the system. By Newton’s law of action and reaction, constraint loads act perpendicular to contact surfaces (if there occurs no friction in kinematic joints) and occur in pairs of equal magnitude and opposite direction. A so-called virtual displacement or kinematically admissible displacement of a system refers to a change in the configuration of the system as the result of any arbitrary infinitesimal change of the coordinates \( q \), consistent with the loads and the constraints imposed on the system at the given instant \( t \). Thus restricting attention to virtual displacements, the total virtual work of all the constraint loads in the system is zero, i.e., \( \delta W = \delta q^T g^c = 0 \), the so-called principle of virtual work. As a result, the constrained variational equations of motion may now be written as

\[ \delta q^T [M\dot{q} - g^a] = 0, \quad (2.4) \]

for all virtual displacements \( \delta q \). The latter equation is the so-called d’Alembert’s principle of virtual work. We have achieved that the constraint loads no longer appear, and the superscript " can now be dropped without ambiguity.

At this point, the well-known classification of constraints into holonomic and nonholonomic becomes important. A constraint is called holonomic if the constraint equations can be expressed as equations connecting the coordinates of the bodies in the following form

\[ \phi(q, t) = 0, \quad (2.5) \]

whereas the nonholonomic constraints can be expressed as

\[ \psi(q, \dot{q}, t) = 0, \quad (2.6) \]
i.e. they depend on \( \dot{q} \) explicitly. For these nonholonomic constraints it is, by definition, impossible to set up a number of equations connecting the coordinates and the time like equation (2.5), since otherwise they would represent holonomic constraints. The simplest examples of nonholonomic systems occur in problems dealing with the rolling motion of one body upon another. In addition, nonholonomic constraints naturally occur during the stick phase of systems exhibiting Coulomb friction (see Section 7.3). In most practical problems \( \psi \) is a linear function of generalized velocities so that equation (2.6) can be written in the form

\[
\psi(q, q, t) = P(q, t) \dot{q} + p(q, t) = 0. \tag{2.7}
\]

For holonomic constraints, the variation of \( \phi \) caused by a variation of the generalized coordinates (with time frozen) is zero, whence we conclude

\[
\phi, \delta q = 0, \tag{2.8}
\]

where \( \phi, := \frac{d\phi}{dq} \) is the Jacobian of \( \phi \) with respect to \( q \). From the nonholonomic constraints (2.7) it follows (cf. [84]) that \( \delta q \) should satisfy the relationship

\[
P\delta q = 0. \tag{2.9}
\]

Hence, equation (2.4) should hold for all virtual displacements \( \delta q \) satisfying the equations (2.8) and (2.9).

Applying the method of Lagrange multipliers (cf. [45, 84]) to equation (2.4) results in

\[
\delta q^T [M \ddot{q} - \dot{g} + H^T \lambda] = 0, \tag{2.10}
\]

for arbitrary \( \delta q \). In this equation \( \lambda \) contains the unknown Lagrange multipliers, accounting for the unknown constraint loads, and \( H := \begin{bmatrix} g, \nu \end{bmatrix}^T \). Since \( \delta q \) is arbitrary in (2.10), we obtain the Lagrange multiplier form of the equations of motion

\[
M \ddot{q} + H^T \lambda = g.
\]

In the remainder of this chapter, however, we will consider holonomic constraints only. So, the equations of motion yield

\[
M \ddot{q} + \phi, \dot{\lambda} = g. \tag{2.11}
\]

### 2.1.3 Augmentation Method

Applying the approach of the previous Subsection, we derived the Lagrange multiplier form of the equations of motion (2.11). In \( \mathbb{R}^n \) this system represents a set of \( 6n \) equations in the \( 6n \) unknown generalized coordinates \( q \) and the \( n_h \) unknown Lagrange multipliers \( \lambda \), where \( n_h \) is the number of holonomic constraints.
2.1. Basic Concepts for Generation of Equations of Motion

In order to describe the motions of the complete system, equation (2.11) has to be augmented by the position constraints (2.5), the velocity constraints and the acceleration constraints. The velocity constraints are obtained by differentiating the position constraints (2.5) with respect to time, giving

$$\phi_q q = -\phi_i = v.$$  \hspace{1cm} (2.12)

Differentiating the position constraints once more one obtains the acceleration constraints

$$\phi_q q = -\phi_q q_i q - 2\phi_q q - \phi_i = \gamma.$$ \hspace{1cm} (2.13)

The equations (2.12) and (2.13) together with (2.11) and (2.5) comprise the complete set of constrained equations of motion for the system. Combining equations (2.11) and (2.13) results in

$$\begin{bmatrix} M & \phi_q \cr \phi_q & 0 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} g \\ \gamma \end{bmatrix}. \hspace{1cm} (2.14)$$

Since, the mass matrix $M$ is positive definite, this system has a unique solution for the accelerations and the Lagrange multipliers if $\phi_q$ has full row rank. Hence, the equations of motion are solvable if the constraint equations are independent. The above method of augmenting the dynamic equations with the constraint equations is called the augmentation method.

This method keeps the unknown constraint forces, taken into account by the Lagrange multipliers, in the equations of motion. Augmenting equation (2.11) by one of the constraint equations (2.5), (2.12) or (2.13) gives a set of DAEs. This method yields the so-called descriptor form of the mechanical system (see [57]). These equations are called Lagrange's equations of type one and are an often used basis for multibody formalisms, e.g. [46, 63, 64].

Example 2.2 The kinematic constraint equations for the pendulum of Example 2.1 read

$$\phi(q) = \begin{bmatrix} x - l \cos \varphi \\ y - l \sin \varphi \end{bmatrix} = 0,$$

where we obtain the generalized coordinate vector $q = [x, y, \varphi]^T$. The velocity and acceleration constraints are

$$\phi_q q = \begin{bmatrix} \dot{x} + l \dot{\varphi} \sin \varphi \\ \dot{y} - l \dot{\varphi} \cos \varphi \end{bmatrix} = 0$$

and

$$\phi_q q = \begin{bmatrix} \dot{x} + l \dot{\varphi} \sin \varphi \\ \dot{y} - l \dot{\varphi} \cos \varphi \end{bmatrix} = -\phi_{qq} q \dot{q} = -\begin{bmatrix} l \dot{\varphi} \cos \varphi \\ l \dot{\varphi} \sin \varphi \end{bmatrix}.$$  

Applying the method of Lagrange multipliers, we obtain

$$\begin{bmatrix} m & m \\ m & J \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} mg \\ J \sin \varphi - l \cos \varphi \end{bmatrix}.$$  

Together, these last two equations form the system of equations (2.14). \hfill \Box
This approach has the advantage of generality and results in relatively simple equations, which can be generated efficiently. The disadvantages are the description by a set of DAEs and the large number of descriptive equations. Commercial codes like ADAMS (cf. [15]); or DADS (cf. [50]) use this method to generate the system equations.

2.1.4 Elimination Method

Another route for deriving the equations of motion is the elimination method. In this approach the system motion is represented by a minimal set of relative variables, called the state variables. The goal of this method is to eliminate both the constraint equations and the constraint loads. Therefore, the system equations are reduced to a so-called state space representation (cf. [73]), i.e., a set of ODEs. The constraint loads may then be computed from a second set of equations. We discuss this method briefly below.

The kinematics of a system of bodies is described by a set of relative coordinates. For that reason, the motion of a particular body in the system is defined with respect to an adjacent body, which motion has previously been defined according to the topological ordering of all bodies in the system. Note, that in the augmentation method one uses a global description, i.e., motions of all bodies in the system are represented with respect to an inertial coordinate frame.

Consider two bodies \( B \) and \( B' \) interconnected by the joint with index \( d \), say. As a result, the relative motion of the two bodies is constrained by \( n_d \) constraint equations. On that ground, the number of degrees of freedom of motion of the bodies relative to each other is

\[
    n_f = \begin{cases} 
    6 - n_d, & \text{in } \mathbb{R}^3, \\
    3 - n_d, & \text{in } \mathbb{R}^2. 
    \end{cases}
\]

So, the motion of the bodies relative to each other can be described by independent relative joint coordinates, denoted by \( q_0, q_d \in \mathbb{R}^n \). These coordinates are required to satisfy the constraint equations automatically. Applying the variational equations of motion (cf. Subsection 2.1.2) for these two bodies, a virtual variation \( \delta q \) may not violate the kinematic constraints between bodies \( B \) and \( B' \). Since relative joint coordinates automatically satisfy the constraints, \( \delta q \) in equation (2.4) is arbitrary. So, it is not necessary to introduce Lagrange multipliers as in equation (2.10). We shall elaborate this in Section 2.2.3.

Example 2.3 The pendulum of example 2.1 has only one degree of freedom, because it is a planar system. Therefore, its motion can be described as a function of the state space variable \( \varphi \). Variation of the generalized coordinates \( q \) gives

\[
    \delta q = \begin{bmatrix} 
    \delta x \\
    \delta y \\
    \delta \varphi 
    \end{bmatrix} = \begin{bmatrix} 
    -l \sin \varphi \\
    l \cos \varphi \\
    1 
    \end{bmatrix} \delta \varphi.
\]
2.2 Recursive Formulation for Constrained Mechanical Systems

Using this expression for the virtual displacements in the principle of virtual work (2.4), we obtain the dynamic equations for the pendulum

\[ \frac{1}{2} ml^2 \ddot{\varphi} + mg \sin \varphi = 0, \]

where the constraint forces have been eliminated.

This procedure can be repeated for the entire multibody system. For so-called open loop systems, i.e. systems in which all bodies are connected in a unique way, this results in a system of ODEs. In Figure 2.2 an open loop system is shown with ten bodies linked together. However, difficulties occur when closed loops appear in the system. In that case

the relative (joint) coordinates are not independent anymore, because two bodies can be connected in more than one way. Wittenburg (cf. [84]) developed the idea of opening the closed loop, by cutting chosen joints, to form an open loop system. The opened system is called the reduced system. First, only this reduced system will be considered. Later, all constraints and all constraint loads which have been ignored in the process of generating the reduced system have to be re-introduced. The now re-introduced constraint loads are provided by adding Lagrange multipliers to the equations of motion. In this way, the original closed loop system is reformulated. Relative description makes recursive formulation possible (cf. [5, 6, 73]). Within the context of multibody dynamics, a recursive formulation is a procedure in which elementary relationships, going for an arbitrary pair of contiguous bodies as part of a system of bodies, can be used all along the system. This can be used to generate the kinematic equations and the system matrices in the dynamic equations in a very efficient way.

2.2 Recursive Formulation for Constrained Mechanical Systems

In this section, a recursive formulation of the equations of motion of spatially constrained mechanical systems is derived. Before explaining the recursive method, we introduce in Subsection 2.2.1 the concept of body connection arrays for describing the way the rigid bodies are linked together by joints. Kinematics of an elementary system of two bodies coupled by an arbitrary joint is discussed in Subsection 2.2.2. The motion of one body
is expressed in terms of the motion of the adjacent body and the relative motion between
the bodies due to the joint. Afterwards, in Subsection 2.2.3, the dynamics for a single
chain, consisting of such elementary systems, is derived. Using a variational form of the
dynamic equations, contributions to the mass matrices and load vectors of the lower num-
bered bodies from all the connected higher numbered bodies are derived. This procedure
is repeated for general open loop systems in Subsection 2.2.3 and a recursive algorithm
is developed to reduce the equations of motion to a base body. In Subsection 2.2.4 this
algorithm is extended to closed loop systems.

The work of Bae and Haug (cf. [5, 6]) is the basis for this section. Roberson and
Schwerdtassek (cf. [73]) obtained the same relations, however, their derivation is different.
They applied the concept of modes of motion and corresponding generalized loads, while
Bae and Haug used the variational form of the system equations.

### 2.2.1 Topology

A multibody system can be very complex and built up of many bodies and many joints.
To create a computer oriented multibody formalism, one must devise a data structure
describing the system topology. One can introduce a so-called system graph (see [84]),
which represents the connectivity of the mechanical system. However, the use of body
connection arrays (cf. [50]) provides a more elegant way to describe the topological rep-
resentation of systems with a tree structure.

Consider a typical tree structured multibody system shown in Figure 2.2, more in
particular the graph whose nodes correspond to the bodies and whose edges represent
the joints. Body $B_1$ is designated as the base body. Formally, the inertial space is intro-
duced as body $B_0$. The remaining bodies are numbered in ascending order along each
tree. Consider the bodies 2 and 3 shown in Figure 2.2; we shall then call node 2 the fa-
thor of node 3. Every body then has a unique father in the tree. The body connection
array $\beta$ is defined by

$$\beta_i := i, \quad \text{where } i = \text{father}(j).$$

So, this array contains the numbers of the connected lower-numbered bodies. This im-
plies that the number of components of $\beta$ equals the number of bodies in the multibody
system. For the system of Figure 2.2 we obtain the connection array

$$\beta = [0, 1, 2, 3, 4, 5, 6, 5, 3, 9]^T.$$  

Joints are numbered such that a joint connecting a father and a son has the number of the
son.

There are two fundamentally different types of mechanical systems. One is an open
loop system (cf. Figure 2.2) and the other is a closed loop system (cf. Figure 2.3). If
a graph has no closed loops, it is called a tree structure. In dealing with closed loops,
the concept of a spanning tree (cf. [73]) is important. A spanning tree is a subgraph of
the system graph which is a tree graph (cf. Figure 2.3) and includes all the nodes of the
original graph. A spanning tree is produced by cutting joints of the multibody system.
2.2. Recursive Formulation for Constrained Mechanical Systems

![Diagram of a closed loop system with a cut joint between bodies $B_n$ and $B_{n+1}$](image)

*Figure 2.3* A closed loop system with a cut joint between bodies $B_n$ and $B_{n+1}$.

graph (cf. Figure 2.3), thereby opening the closed loop system and forming the reduced system, as already explained in Subsection 2.1.4. This implies the division of joints into tree structure joints or primary joints, defining the spanning tree, and into cut joints or secondary joints that close the multibody system.

If a node is connected to more than two edges, it is called a *junction node*. Nodes 3 and 5 in Figure 2.2 are junction nodes for example. The general definition of a *chain* in graph theory is a path between nodes. However, here the term chain will be used only for paths between junction nodes or between a junction node and a tree end node. In the closed loop system shown in Figure 2.3 the junction node is denoted by $B_i$. Cutting the joint between the bodies $B_i$ and $B_{i+1}$ defines the chains $\gamma_1 = B_i$ ($i = l, l + 1, \ldots, n$) and $\gamma_2 = B_i$ ($i = l, m, m - 1, \ldots, n + 1$). In a spanning tree there is a unique path along the graph of the system from the base body to each body in the system.

### 2.2.2 Kinematic Relations

Before describing a recursive formulation for setting up the equations of motion for multibody systems, the kinematics of these systems is discussed. To derive our kinematic relations, we need to examine a pair of adjacent bodies as shown in Figure 2.4.

![Diagram of two bodies $B_i$ and $B_j$ interconnected by an arbitrary joint $j$, since $i =$ father($j$). The absolute positions of the centroids of the bodies $B_i$ and $B_j$ are given by the position vectors $r_i$ and $r_j$, respectively. For the joint $j$ the locations of the joint attachment points on $B_i$ and $B_j$ are defined by $s_{ij}$ and $s_{ij}$, respectively. Usually an interconnection constrains the motion. For each joint, depending on the degrees of freedom, relative generalized coordinates can be defined to describe the relative motion of the linked bodies. For example, in a translational joint the elongation along the translational axis can be used as a relative](image)

*Figure 2.4* A pair of linked bodies.

two bodies $B_i$ and $B_j$ interconnected by an arbitrary joint $j$, since $i =$ father($j$). The absolute positions of the centroids of the bodies $B_i$ and $B_j$ are given by the position vectors $r_i$ and $r_j$, respectively. For the joint $j$ the locations of the joint attachment points on $B_i$ and $B_j$ are defined by $s_{ij}$ and $s_{ij}$, respectively. Usually an interconnection constrains the motion. For each joint, depending on the degrees of freedom, relative generalized coordinates can be defined to describe the relative motion of the linked bodies. For example, in a translational joint the elongation along the translational axis can be used as a relative...
generalized coordinate, and in Example 2.3 we have used the angle \( \varphi \) as a relative generalized coordinate. From Figure 2.4, the following relationships between the absolute position vectors of body \( B_j \) and body \( B_i \) can be obtained

\[
\mathbf{r}_{ij} = \mathbf{r}_i + \mathbf{r}_j,
\]

(2.16)

where

\[
\mathbf{r}_{ij} = \mathbf{s}_{ij} + \mathbf{d}_j - \mathbf{s}_{ji},
\]

(2.17)

and, where the joint displacement vector is a function of the generalized coordinates, i.e. \( \mathbf{d}_j = \mathbf{d}_j(q_j) \). We derive relationships between the velocities in the system. The absolute angular velocity \( \omega_{ij} \) of body \( B_j \) can be expressed in terms of the absolute angular velocity \( \omega_i \) of body \( B_i \) and the relative angular velocity \( \omega_{ij} \) between bodies \( B_i \) and \( B_j \) by the following relation

\[
\omega_{ij} = \omega_i + \omega_{ij} = \omega_i + \mathbf{H}_{ij} \dot{q}_j,
\]

(2.18)

since the relative angular velocity \( \omega_{ij} \) can be expressed in terms of the relative generalized velocities by the linear relation \( \omega_{ij} = \mathbf{H}_{ij} \dot{q}_j \) (see [45]). We obtain for the time derivatives of \( s_{ij} \)

\[
\dot{s}_{ij} = \omega_i \times s_{ij},
\]

(2.19)

and likewise for \( s_{ji} \). Using equation (2.18) we find

\[
\dot{s}_{ij} = (\omega_i + \mathbf{H}_{ij} \dot{q}_j) \times s_{ij}.
\]

(2.20)

The derivative of \( \mathbf{d}_j \) can be obtained as a combination of the absolute change of its orientation and its relative velocity, viz.

\[
\dot{\mathbf{d}}_j = \dot{\omega}_i \times \mathbf{d}_j + \frac{\partial \mathbf{d}_j}{\partial q_j} \dot{q}_j,
\]

(2.21)

where \( \frac{\partial \mathbf{d}_j}{\partial q_j} \) is the Jacobian matrix of the joint vector \( \mathbf{d}_j \) with respect to the relative joint coordinates \( q_j \). As a result we find

\[
\dot{r}_j = \dot{r}_i + \omega_i \times r_{ij} - (\mathbf{H}_{ij} \dot{q}_j) \times s_{ij} + \frac{\partial \mathbf{d}_j}{\partial q_j} \dot{q}_j
\]

\[
= \dot{r}_i - \dot{\mathbf{R}}_i, \omega_i + (\dot{S}_{ij} \mathbf{H}_j + \frac{\partial \mathbf{d}_j}{\partial q_j}) \dot{q}_j,
\]

(2.22)

where a skew-symmetric matrix \( \dot{\mathbf{Z}} \) with axial vector \( \mathbf{z} \) is defined by \( \mathbf{z} \times \mathbf{a} = \dot{\mathbf{Z}} \mathbf{a} \), \( \forall \mathbf{a} \). Note that the first two right-hand terms of equation (2.22) correspond to absolute velocity terms, whereas the last term is expressed in terms of the time derivatives of the relative generalized coordinates of joint \( j \). Equations (2.18) and (2.22) are the kinematic
2.2. Recursive Formulation for Constrained Mechanical Systems

Equations relating absolute velocities and relative velocities. Defining $\mathbf{v}_i := \begin{bmatrix} \dot{\mathbf{x}}_i \\ \dot{\mathbf{r}}_i \end{bmatrix}$, equations (2.18) and (2.22) can be composed as

$$\mathbf{v}_j = \mathbf{A}_j \mathbf{v}_i + \mathbf{B}_j \mathbf{q}_j,$$

where

$$\mathbf{A}_j := \begin{bmatrix} \mathbf{I} & -\mathbf{R}_j \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_j := \begin{bmatrix} \mathbf{S}_{\hat{\mathbf{H}}_j} + \frac{\partial \mathbf{S}}{\partial \mathbf{q}_j} \\ \mathbf{H}_j \end{bmatrix}.$$  \hspace{1cm} (2.24)

For the derivation of the dynamic equations in Subsection 2.2.3 we will use the variational form of the system equations, for which we need expressions for the virtual displacements. Analogously to the derivation of the velocity equation (2.23) the virtual displacement equation (2.25) can be derived by replacing the time derivative by the variation operator (see [45]), i.e.

$$\delta \mathbf{u}_j = \mathbf{A}_j \delta \mathbf{u}_i + \mathbf{B}_j \delta \mathbf{q}_j,$$  \hspace{1cm} (2.25)

with

$$\delta \mathbf{u}_i := \begin{bmatrix} \delta \mathbf{r}_i \\ \delta \mathbf{\pi}_i \end{bmatrix},$$  \hspace{1cm} (2.26)

where $\delta \mathbf{r}_i$ and $\delta \mathbf{\pi}_i$ denote virtual translation and virtual rotation, respectively, and $\delta \mathbf{q}_j$ denotes virtual relative coordinates.

The matrix form of acceleration relations can be obtained by differentiation of equation (2.23), i.e.

$$\ddot{\mathbf{v}}_j = \mathbf{A}_j \dot{\mathbf{v}}_i + \mathbf{B}_j \dot{\mathbf{q}}_j + \mathbf{c}_j,$$  \hspace{1cm} (2.27)

where

$$\mathbf{c}_j := \dot{\mathbf{A}}_j \mathbf{v}_i + \dot{\mathbf{B}}_j \mathbf{q}_j.$$  \hspace{1cm} (2.28)

$$\dot{\mathbf{A}}_j = \begin{bmatrix} \mathbf{O} & -\dot{\mathbf{R}}_j \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{B}}_j := \begin{bmatrix} \dot{\mathbf{S}}_{\hat{\mathbf{H}}_j} + \mathbf{S}_{\frac{\partial \mathbf{H}_j}{\partial \mathbf{q}_j}} \\ \frac{\dot{\mathbf{H}}_j}{\dot{\mathbf{q}}_j} \end{bmatrix}.$$  \hspace{1cm} (2.29)

The matrices $\mathbf{A}_j$ and $\dot{\mathbf{A}}_j$ are independent of joint type, while $\mathbf{B}_j$ and $\mathbf{B}_j$ depend upon the joint type and its relative coordinates. Equations (2.23), (2.25) and (2.27) are the recursive kinematic relations. The method described above shall be demonstrated on a simple example.

**Example 2.4** Consider a compound pendulum shown in Figure 1.4, where the bodies $B_1$ and $B_2$ have lengths $2l_1$ and $2l_2$, respectively. The relative joint coordinates are $\mathbf{q}_1 = \varphi_1$ and $\mathbf{q}_2 = \varphi_2$. The position of the centre of mass of rod $B_1$ is given by

$$\mathbf{r}_1 = \begin{bmatrix} l_1 \cos \varphi_1 \\ l_1 \sin \varphi_1 \end{bmatrix}.$$  \hspace{1cm} (2.30)
The position of the centroid of rod \( B_2 \) can be expressed as
\[
\mathbf{r}_2 = \mathbf{r}_1 + \mathbf{r}_{12} = \begin{bmatrix}
2l_1 \cos \phi_1 + l_2 \cos(\phi_1 + \phi_2) \\
2l_1 \sin \phi_1 + l_2 \sin(\phi_1 + \phi_2)
\end{bmatrix}.
\] (2.31)

Denote the inertial frame as body \( B_0 \), implying that \( \mathbf{v}_0 = 0 \) and \( \omega_0 = \mathbf{0} \). Since \( \mathbf{H}_1 = 1 \), \( \mathbf{d}_1 = \mathbf{0} \) and \( \mathbf{a}_1 = \begin{bmatrix} -l_1 \cos \phi_1 \\
l_1 \sin \phi_1 \\
1 \end{bmatrix} \), we find
\[
\mathbf{B}_1 = \begin{bmatrix}
-l_1 \sin \phi_1 \\
l_1 \cos \phi_1 \\
1
\end{bmatrix}.
\]

Therefore, for rod \( B_1 \) the following kinematic relations hold
\[
\mathbf{v}_1 = \mathbf{B}_1 \dot{\mathbf{q}}_1, \quad \mathbf{v}_1 = \mathbf{B}_1 \dot{\mathbf{q}}_1 + \mathbf{B}_1 \dot{\mathbf{q}}_1 \quad \text{and} \quad \delta \mathbf{u}_1 = \mathbf{B}_1 \delta \mathbf{q}_1.
\] (2.32)

For \( \mathbf{A}_2 \) and \( \mathbf{B}_2 \) we find
\[
\mathbf{A}_2 = \begin{bmatrix}
1 & 0 & -l_1 \sin \phi_1 - l_2 \sin(\phi_1 + \phi_2) \\
0 & 1 & l_1 \cos \phi_1 + l_2 \cos(\phi_1 + \phi_2) \\
0 & 0 & 1
\end{bmatrix} \quad \text{and} \quad \mathbf{B}_2 = \begin{bmatrix}
-l_2 \sin(\phi_1 + \phi_2) \\
l_2 \cos(\phi_1 + \phi_2) \\
1
\end{bmatrix}.
\]

Therefore, the kinematic relations for velocities, accelerations and virtual displacements of \( B_2 \) are
\[
\mathbf{v}_2 = \mathbf{A}_2 \mathbf{v}_1 + \mathbf{B}_2 \dot{\mathbf{q}}_2, \quad \mathbf{v}_2 = \mathbf{A}_2 \mathbf{v}_1 + \mathbf{B}_2 \dot{\mathbf{q}}_2 + \mathbf{c}_2 \quad \text{and} \quad \delta \mathbf{u}_2 = \mathbf{A}_2 \delta \mathbf{u}_1 + \mathbf{B}_2 \delta \mathbf{q}_2.
\] (2.33)

respectively. Note that differentiating (2.30) and (2.31) with respect to time and augmenting them with the rotational velocities results in the same relations.

### 2.2.3 Open Loop Systems

In practice, open loop multibody systems appear less frequently than systems with closed chains. However, there are two reasons to treat this class of systems first. One reason is the greater simplicity of the mathematical description of the interconnection structure and of the kinematics. The second reason is that, by cutting joints, the equations of motion for a system with closed chains can be obtained from the reduced system by introduction of Lagrange multipliers as explained in Subsection 2.1.4. Cutting joints results in a tree structure mechanism only consisting of chains. We employ the recursive formulation, developed in Subsection 2.2.2, to a typical chain \( \gamma \) (cf. Figure 2.5), beginning at a junction node \( l \) and proceeding to a tree end node \( n \). The variational equations of motion for chain \( \gamma \) are reduced to equations for a single body, viz. the junction body \( B_l \). The variational equations of motion for the whole chain \( \gamma \) (cf. (2.4)) are
\[
\sum_{i \in \gamma} \delta \mathbf{u}_i^\top [\mathbf{M}_i \dot{\mathbf{q}}_i - \mathbf{g}_i] = 0,
\] (2.34)
where $\delta u_i$, $i = l, l+1, \ldots, n$, are kinematically admissible for the constraints between the bodies $B_l$ to $B_n$, and any other constraints that may act on the junction body $B_l$. Note that $M_l$ is defined as in Subsection 2.2.1. Repeated usage of the kinematic relations (2.25) and (2.27) between bodies the $B_l$ and $B_{l+1}$ together with the joint coordinates $q_j$, since $j-1 = \text{father}(j)$, leads to the recursive equations of motion for $B_l$ (see Appendix A for a detailed derivation)

$$\delta u_l^T [(M_l + \tilde{M}_l)v_l - (g_l + \tilde{g}_l)] = 0. \quad (2.35)$$

Here, $\delta u_l$ is kinematically admissible for all external constraints that act on the junction body $B_l$. The recursive mass matrix $\tilde{M}_l$ and the recursive load vector $\tilde{g}_l$ for chain $\gamma$ are defined in Appendix A.

For more general open loop systems this procedure can be repeated for each chain (see [81] for an elaboration). The recursive mass matrices and load vectors of all chains, originating from the same junction body, are added together. The successive reduction process can be repeated all along the rest of the tree structure until one reaches the base body $B_l$. It results in the variational equations of motion for $B_l$, i.e.

$$\delta u_l^T [(M_l + \tilde{M}_l)v_l - (g_l + \tilde{g}_l)] = 0, \quad (2.36)$$

for the whole tree $\gamma$. Now, $\delta u_l$ is kinematically admissible for the kinematic constraint between the base body $B_l$ and inertial space, regarded as $B_0$ (see Subsection 2.2.1). Since $B_0$ has constant position, the position of body $B_l$ relative to inertial space can be represented by joint coordinates between $B_l$ and $B_0$, just like any other arbitrary joint between two adjacent bodies $B_l$ and $B_{l+1}$. The advantage of this procedure (cf. [75]) is that the introduction of Lagrange multipliers (as in [5]) is not needed. Therefore, it provides the advantage of maintaining a system of ODEs and avoids having to solve a system of DAEs as in [5].

The basic advantage of this process is the fact that the dynamic equations are generated in explicit form (see equations (A.2) and (A.6) in Appendix A) with a number of operations, which increases only linearly with the number $n$ of the system bodies. Therefore, they are called $O(n)$-formulations (cf. [74]). Besides, the small dimension of the matrices that have to be inverted to obtain these joint accelerations (cf. (A.2) and (A.6)) is advantageous. For open loop systems this reduction process can be explained by the following

Example 2.5 For the double pendulum of Figure 1.4 the unreduced equations of motion read

$$\delta u_l^T [(M_l v_l - g_l) + \delta u_l^T (M_{l+1} v_{l+1} - g_{l+1})] = 0, \quad (2.37)$$
where
\[
\mathbf{M}_i = \begin{bmatrix} m_i & m_i \phi_i \\ m_i \phi_i & J_i \end{bmatrix} \quad \text{and} \quad \mathbf{g}_i = \begin{bmatrix} m_i g \\ 0 \\ 0 \end{bmatrix}.
\]

Substitution of (2.33) into (2.37) results in
\[
\delta \mathbf{u}_i^T \left( \mathbf{M}_i \dot{\mathbf{v}}_i - \mathbf{g}_i \right) + \left[ \delta \mathbf{u}_i^T \mathbf{A}_i^T + \delta \mathbf{u}_i^T \mathbf{B}_i^T \right] \left( \mathbf{M}_i \phi_i \dot{\mathbf{v}}_i + \mathbf{B}_i \phi_i \mathbf{q}_i \mathbf{q}_i - \mathbf{c}_i \right) - \mathbf{g}_i = 0.
\]

Since \( \delta \mathbf{q}_i \) is arbitrary, we find
\[
\dot{\mathbf{q}}_i = \dot{\mathbf{v}}_i = - \left( \mathbf{B}_i^T \mathbf{M}_i \mathbf{B}_i \right)^{-1} \mathbf{B}_i^T \left( \mathbf{M}_i \phi_i \dot{\mathbf{v}}_i + \mathbf{B}_i \phi_i \mathbf{q}_i \mathbf{q}_i - \mathbf{c}_i \right) - \mathbf{g}_i.
\] (2.38)

This can be rewritten as
\[
(m_i \phi_i^2 + J_i) \left( \dot{\phi}_i + \dot{\phi}_i \right) + 2m_i l_i l_i \left( \dot{\phi}_i \cos \phi_2 - \dot{\phi}_i \sin \phi_2 \right) + m_i g l_i \sin \left( \phi_1 + \phi_2 \right) = 0.
\]

Substitution of (2.38) into the equations of motion gives
\[
\delta \mathbf{u}_i^T \left( \left[ \mathbf{M}_i + \mathbf{M}_i \right] \dot{\mathbf{v}}_i - \left( \mathbf{g}_i + \mathbf{g}_i \right) \right) = 0.
\] (2.39)

Using the kinematic relations (2.32) in equation (2.39) results in
\[
\delta \mathbf{u}_i^T \left[ \left( \mathbf{M}_i + \mathbf{M}_i \right) \left( \mathbf{B}_i \dot{\mathbf{q}}_i + \mathbf{B}_i \mathbf{q}_i \right) - \left( \mathbf{g}_i + \mathbf{g}_i \right) \right] = 0,
\]

where \( \delta \mathbf{q}_i \) is arbitrary. Therefore, one finds
\[
\dot{\mathbf{q}}_i = \dot{\mathbf{v}}_i = - \left( \mathbf{B}_i^T \left[ \left( \mathbf{M}_i + \mathbf{M}_i \right) \right] \mathbf{B}_i \right)^{-1} \mathbf{B}_i^T \left( \left[ \mathbf{M}_i + \mathbf{M}_i \right] \left( \mathbf{B}_i \dot{\mathbf{q}}_i + \mathbf{B}_i \mathbf{q}_i \right) - \left( \mathbf{g}_i + \mathbf{g}_i \right) \right).
\] (2.40)

Equations (2.38) and (2.40) give the joint accelerations of \( B_2 \) and \( B_1 \), respectively, in explicit form.

### 2.2.4 Closed Loop Systems

Most multibody systems found in practice do not have a tree structure, but consist of closed loops. In the Subsections 2.1.4 and 2.2.1 we have already mentioned that these systems are quite different from open loop systems. In setting up the equations of motion for such systems the results of previous subsections can be used.

In Figure 2.3 a closed loop system is shown. There, body \( R_i \) is the junction body. As explained in Subsection 2.2.1 a spanning tree can be obtained by cutting the joint between two chosen bodies \( R_i \) and \( R_{i+1} \), say. Two chains, denoted \( \gamma_i \) and \( \gamma_{i+1} \), are defined thereby. For these chains the variational equations of motion are reduced. The equations of motion for the unreduced system are
\[
\sum_{i \neq n, n+1} \delta \mathbf{u}_i^T \mathbf{M}_i \dot{\mathbf{v}}_i - \mathbf{g}_i + \sum_{i \neq n} \delta \mathbf{u}_i^T \mathbf{M}_i \dot{\mathbf{v}}_i - \mathbf{g}_i - \phi_{i+1} = 0.
\] (2.41)
2.2. Recursive Formulation for Constrained Mechanical Systems

where \( \delta u_t \) are kinematically admissible for the tree structure constraints. The motion of the bodies must satisfy \( \phi = 0 \), representing the constraint for the cut joint between bodies \( B_t \) and \( B_{t+1} \). As in Subsection 2.1.2, the Lagrange multipliers \( \lambda \) account for the constraint loads for this cut joint. As explained in the procedure of Appendix B the equations of motion for the closed loop system can be reduced to

\[
\delta u_t^T [ (M_t + \tilde{M}_t) + \tilde{M}'_t ) \dot{v}_t - (g_t + \tilde{g}_t + \tilde{g}'_t ) \] + \left[ \tilde{\Theta}^T_t + \tilde{\Theta}'^T_t \right] \lambda = 0.
\] (2.42)

Here, \( \delta u^T_t \) is kinematically admissible for all constraints that act on body \( B_t \), other than those associated with the chains \( \gamma_1 \) and \( \gamma_2 \) in Figure 2.3. Resulting from the recursive elimination along each chain \( \gamma_i \), \( i = 1, 2 \), \( \tilde{M}'_t \), \( \tilde{g}'_t \) and \( \tilde{\Theta}'_t \) (cf. Appendix B) denote the recursive mass, the recursive load and the recursive 

As in Subsection 2.1.3, equation (2.42) (cf. (2.4)) has to be augmented by the constraint equations, resulting in a system of DAEs. Here, the constraint equations result from the cut joint constraint between the bodies \( B_t \) and \( B_{t+1} \). Let the vector \( p_t \) denote the so-called rotational degrees of freedom \( \gamma \) for body \( B_t \). Then the cut constraint reads

\[
\phi(u_t, u_{t+1}, \gamma) = 0,
\] (2.43)

where \( u_t := [v_t^T] \), likewise for \( u_{t+1} \). As in Subsection 2.1.3 the velocity constraints with respect to the cut are

\[
\dot{\phi} = \phi_{u_t} v_t + \phi_{u_{t+1}} v_{t+1} - \nu = 0,
\] (2.44)

and for the acceleration constraints for the cut joint one finds

\[
\ddot{\phi} = \phi_{u_t} \ddot{v}_t + \phi_{u_{t+1}} \ddot{v}_{t+1} - \gamma = 0.
\] (2.45)

Basically, combining equation (2.41) with any of the equations (2.43), (2.44) or (2.45) results in a solvable system of equations. The recursive technique as described in the previous subsections may be applied to the cut constraints (2.43), (2.44) and (2.45), resulting in constraints for the base body. These reduced constraints together with (2.42) yield a solvable system.

Consider the cut acceleration constraints (2.45) for example. Using the kinematic relationships (2.23) and (2.27) and expression (B.6) in Appendix B, we proceed as before to eliminate recursively the intermediate bodies down to the base body. This results in

\[
(\tilde{\Theta}^T_t + \tilde{\Theta}'^T_t) \dot{v}_t + (\tilde{\Theta}^T_t + \tilde{\Theta}'^T_t) \lambda = \ddot{\gamma}_t + \dot{\gamma}_t + \gamma.
\] (2.46)

Here, the recursive Jacobian \( \tilde{\Theta} \), the recursive \( \tilde{\Theta}' \) and the recursive right-hand term \( \hat{\gamma} \) are defined by

\[
\tilde{\Theta}^T_t = A_{t+1}^T [ I - (M_{t+1} + \tilde{M}_t)B_{t+1} (B_{t+1}^T (M_{t+1} + \tilde{M}_t)B_{t+1})^{-1} B_{t+1}^T ] \tilde{\Theta}^T_{t+1},
\]
\[
\tilde{\Theta}'_t = \tilde{\Theta}_{t+1} - \tilde{\Theta}_{t+1} B_{t+1} (B_{t+1}^T [M_{t+1} + \tilde{M}_t]B_{t+1})^{-1} B_{t+1}^T \tilde{\Theta}'_{t+1},
\]

Joints having three rotational degrees of freedom are called free rotational joints, e.g. spherical joints. Four Euler parameters are defined as relative orientation generalized coordinates for such joints. Then the normalization constraint \( \| \hat{p} \|_2 = 1 \) has to be satisfied by the Euler parameters.
and
\[ \ddot{Y}_i := \dot{Y}_{i+1} - \vec{\Phi}_{i+1}\dot{\Phi}_{i+1} + \vec{\Phi}_{i+1}\dot{B}_{i+1} (\vec{B}_{i+1}^T (M_{i+1} + \vec{M}_{i+1}) B_{i+1})^{-1} \vec{B}_{i+1}^T \]
\[ \cdot \left( (M_{i+1} + \vec{M}_{i+1}) C_{i+1} - (g_{i+1} + \vec{g}_{i+1}) \right) . \]

Combining equations (2.42) and (2.46) gives a system of equations
\[ \begin{bmatrix} \ddot{Z}_i \\ \vec{\phi}_i \\ \vec{\psi}_i \\ \lambda_i \end{bmatrix} = \begin{bmatrix} \ddot{g}_i \\ \ddot{\phi}_i \\ \ddot{\psi}_i \end{bmatrix}, \tag{2.47} \]
where
\[ \vec{M}_i := M_i + \vec{M}_i^T + \vec{M}_i^\top, \quad \vec{\phi}_i := \vec{\phi}_i^T + \vec{\phi}_i^\top, \quad \vec{\psi}_i := \vec{\psi}_i^T + \vec{\psi}_i^\top, \]
\[ \ddot{g}_i := g_i + \ddot{g}_i^T + \ddot{g}_i^\top \quad \text{and} \quad \ddot{\lambda}_i := \ddot{\lambda}_i^T + \ddot{\lambda}_i^\top + \ddot{\lambda}_i. \]

Equation (2.47) has a unique solution for the accelerations \( \ddot{\psi}_i \) and the Lagrange multipliers \( \lambda_i \) if the cut constraints are independent.

**Example 2.6** Consider the plane crank-slider mechanism shown in Figure 1.5. The mechanism consists of a crank \( B_1 \) of length 2l₁ and a connecting slider \( B_2 \) of length 2l₂. The slider is constrained to move along the x-axis. This multibody system is a closed loop system. We can form a spanning tree by removing the joint between the slider and inertial space. Therefore, we define the cut joint constraint
\[ \phi(u_2, t) = r_{1,2} + l_2 \sin(\phi_1 + \phi_2) = 0, \]
with \( u_2 = [r_{1,2}, r_{2,3}, \phi_2] \). The other kinematic relations are the same as for the double pendulum. For the crank-and-slider mechanism the unreduced equations of motion are
\[ \delta u_1^T (M_1 \dot{\phi}_1 - \dot{g}_1) + \delta u_2^T (M_2 \dot{\phi}_2 - \dot{g}_2 + \phi_2^T \lambda) = 0, \tag{2.48} \]
for kinematically admissible displacements \( \delta u_1 \) and \( \delta u_2 \). In the equation above \( \phi_{\infty} \) is
\[ \phi_{\infty} = [0, 1, l_2 \cos(\phi_1 + \phi_2)]^T. \]

Substitution of the kinematic relations (2.33) into (2.48) gives
\[ \delta u_1^T (M_1 \dot{\phi}_1 - \dot{g}_1) + \delta u_2^T \left[ M_2 (A_2 \dot{\phi}_1 + B_2 \dot{q}_2 + c_2) - \dot{g}_2 + \phi_2^T \lambda \right] + \]
\[ \delta q_1^T B_2^T (M_2 (A_2 \dot{\phi}_1 + B_2 \dot{q}_2 + c_2) - \dot{g}_2 + \phi_2^T \lambda) = 0. \]

Now, \( \delta q_2 \) is arbitrary, and therefore we find
\[ \dot{q}_2 = \dot{\phi}_0 = -B_2^T (M_2 B_2)^{-1} B_2^T (M_2 (A_2 \dot{\phi}_1 + c_2) - \dot{g}_2 + \phi_2^T \lambda), \tag{2.49} \]
with the constraint already mentioned. Substitution of the equation for \( \dot{q}_2 \) in the equations of motion gives
\[ \delta u_1^T (M_1 + \vec{M}_1) \dot{\phi}_1 - (g_1 + \ddot{g}_1) + \vec{\phi}_1^T \lambda) = 0. \]
2.2. Recursive Formulation for Constrained Mechanical Systems

Using the kinematic relations (2.32) between rod $B_i$ and inertial space, the equations of motion result in

$$
\delta q_1 B_i^T[(M_i + \tilde{M}_i)(\dot{\tilde{q}}_1 + c_0) - (g_i + \tilde{g}_1) + \tilde{\Phi}_i^T \lambda] = 0,
$$

where $\delta q_1$ is arbitrary. Therefore,

$$
\ddot{q}_1 = \dot{\tilde{q}}_1 = - (B_i^T(M_i + \tilde{M}_i)B_i)^{-1}B_i^T[(M_i + \tilde{M}_i)c_1 - (g_i + \tilde{g}_1) + \tilde{\Phi}_i^T \lambda].
$$

(2.50)

The constraint yields

$$
\phi(u_2, t) = l_1 \sin \varphi_1 + l_2 \sin(\varphi_1 + \varphi_2) = 0.
$$

(2.51)

Combining equations (2.49) and (2.50) with constraint equation (2.51) results in a set of differential algebraic equations for the unknowns $\varphi_1$, $\varphi_2$ and $\lambda$.

2.2.5 Recursive Algorithm

The results of the previous Subsections 2.2.2, 2.2.3 and 2.2.4 can be summarized as follows. The reduced forms of the kinematic and dynamic equations of motion of a multibody system can be generated recursively by the following scheme:

(i). Positions and velocities of all rigid bodies in the system can be generated by means of equations (2.16), (2.17), and (2.23) ($r_j$, $v_j$, for $j = 1, 2, \ldots, n_b$).

(ii). Generalized loads $g_j$ can be generated for each body in the system, and the generalized mass matrix $M_j$ can be composed for each body ($j = 1, 2, \ldots, n_b$).

(iii). If closed loops are present in the system, a spanning tree has to be formed by cutting joints (cf. Section 2.2.1).

(iv). The reduced variational equations of motion for first level junction nodes, i.e. junction nodes with chains that terminate at tree end bodies or cut-joint bodies, can be obtained by adding contributions from the variational equations of each chain that starts from each junction node (cf. Subsections 2.2.3 and 2.2.4).

(v). Repeating this procedure for every chain by working back to the base body, the reduced variational equations of motion for the base body can be obtained.

(vi). For closed loop systems the equations of motion have to be augmented by the cut constraint equations (2.43), (2.44) and (2.45), in reduced form (2.46).

The equations of motion for open loop systems are ODEs (cf. Subsection 2.2.3). Integration techniques for ODEs are well developed. However, the resulting equations for closed loop systems are DAEs (cf. Subsection 2.2.4). These equations are not differential equations (cf. [65]). The lack of broadly applicable and robust integration methods for DAEs remains the fundamental limitation in automated application and effective use of multibody dynamic simulation methods.
2.3 CONCLUSIONS

The methods for generating multibody system equations have been surveyed here. To describe the motion, one may select either the descriptor form (cf. Subsection 2.1.3) or the state space representation (cf. Subsection 2.1.4 and Section 2.2). The descriptor form is obtained most efficiently in terms of absolute variables, describing the motion of the bodies with respect to inertial space. The resulting equations of motion constitute a set of DAEs.

For open loop systems, the state space form can be generated most efficiently in terms of relative variables, representing the motion of interacting bodies relative to each other, by using recursive formulations (cf. Subsection 2.2.3). A merit of the use of relative generalized coordinates is that they automatically satisfy the kinematic constraints between the bodies. This has the effect that the dynamic equations of the system do not have to be augmented by extra algebraic equations for the constraints. Hence, this method results in a set of ODEs for open loop systems. Recursive formulation generates these equations very efficiently. As already mentioned in Subsection 2.2.3 the main advantage of recursive O(n) formulations is the fact that only small mass matrices have to be inverted, thereby obtaining a high efficiency. Another advantage is the possibility of parallel processing. Parallel processors can be used to do simultaneous independent computations. Computation of recursive inertial and right-hand side terms in different branches of a mechanism are independent, therefore parallel processors can be used. Moreover, parallel computations can be applied to recover body and joint accelerations.

However, for the frequently appearing closed loop systems the dynamic equations have to be augmented by constraint equations, resulting from opening the closed loop by cutting joints. Thus, in the case of closed loop systems the same recursive computational scheme yields a set of DAEs.

Since the dynamic behaviour of most multibody systems is described by DAEs, in the remainder of this thesis we will focus our attention on the analysis and the numerical solution of DAEs.
DIFFERENTIAL ALGEBRAIC EQUATIONS

Differential algebraic equations (DAEs) arise in many applications, like mechanical systems with constraints, modelling of electrical networks and flow of incompressible fluids. This class of problems presents numerical and analytical difficulties which are quite different from ordinary differential equations (ODEs). In this chapter the theory and the numerical solution of DAEs are examined. In Section 3.1 we give a general introduction to DAEs, i.e. differential equations subject to constraints. A theory for linear DAEs is developed. The notion of matrix pencil appears to be crucial for these systems. The concept of the index, which characterizes DAEs, is introduced. Next, nonlinear DAEs are considered. For such systems the notion of index is extended, viz. the differential and the perturbation index. Thereafter semi-explicit systems, which form an important class of DAEs, are introduced. Sections 3.2 and 3.3 deal with the study of numerical methods applied to DAEs; Section 3.2 deals with multistep methods, whereas a brief overview of Runge-Kutta methods is given in Section 3.3.

3.1 THEORY OF DIFFERENTIAL ALGEBRAIC EQUATIONS

3.1.1 Introduction

The general form of a so-called implicit differential equation is given by

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

(3.1)

where \( x : [0, T] \rightarrow \mathbb{R}^n \) and where the function \( f : [0, T] \times \mathbb{R}^{2n} \rightarrow \mathbb{R}^n \) is assumed to be sufficiently differentiable. The Jacobian matrix \( \frac{\partial f}{\partial \dot{x}} \) may be singular. This class of differential equations includes ODEs as a special case. If \( \frac{\partial f}{\partial \dot{x}} \) is nonsingular, equation (3.1) is locally a system of ODEs. However, if the Jacobian is singular, equation (3.1) is in fact a system of DAEs. In such a system there are algebraic constraints on the variables.
3.1.2 Linear DAEs with Constant Coefficients

The simplest and best understood problems of the form (3.1) are linear differential algebraic equations with constant coefficients

\[ \dot{x}(t) + Bx(t) = g(t), \]

where the coefficients \( A, B \in \mathbb{R}^{n \times n} \), and the forcing function \( g : [0, T] \rightarrow \mathbb{R}^n \). We study these systems to gain insight in the behaviour of solutions of DAEs.

Consider the DAE (3.2) with a nonsingular matrix \( A \). This equation can be rewritten as

\[ \dot{x}(t) = -A^{-1}Bx(t) + A^{-1}g(t), \]

which is just a (familiar) explicit ODE. Hence, we concentrate on the case of singular matrices \( A \). If \( g = 0 \) in (3.2) the Ansatz

\[ x(t) = \exp(\lambda t)x_0, \]

leads to the relation

\[ \det(\lambda A + B) = 0. \]

It is therefore useful to consider the matrix pencil \( (A, B) \). This matrix pencil is called singular if \( \lambda A + B \) is singular for all values of \( \lambda \); otherwise it is called regular.

**Definition 3.1** The vector \( x_0 \in \mathbb{R}^n \) is said to be a consistent initial vector if problem (3.2) with the initial value \( x(0) = x_0 \) possesses at least one solution. If the initial value problem possesses a unique solution for all consistent initial vectors, then the problem is called solvable.\(^1\)

Solvability of a linear DAE with constant coefficients can be characterized by the following theorem.

**Theorem 3.2** For \( A, B \in \mathbb{R}^{n \times n} \) the DAE

\[ \dot{x}(t) + Bx(t) = g(t) \]

is solvable if and only if the matrix pencil \( (A, B) \) is regular.

**Proof.** Suppose the matrix pencil \( (A, B) \) is singular. Choose an arbitrary set of \( n + 1 \) distinct \( \lambda_t \) and \( v_t \neq 0 \), \( t = 1, 2, \ldots, n + 1 \) such that \( \lambda_t A + B \) and \( v_t \). There is a nontrivial combination \( \sum_{t=1}^{n+1} \alpha_t v_t = 0 \), but there exists a \( t > 0 \) such that \( \sum_{t=1}^{n+1} \alpha_t \exp(\lambda_t t)v_t(t) \neq 0 \). For that reason, the problem \( A\dot{x} + Bx = 0 \) with \( x(0) = 0 \) has two different solutions, namely \( \sum_{t=1}^{n+1} \alpha_t \exp(\lambda_t t)v_t \) and \( 0 \). Therefore the DAE (3.2) is not solvable for a singular matrix pencil \( (A, B) \).

---

\(^1\) Campbell (cf. [14]) and Griepentrog and März (cf. [40]) use the term *tractable* instead of solvable.
3.1. Theory of Differential Algebraic Equations

On the other hand, assume that the matrix pencil is regular. Then, the polynomial \( p(z) = \det(zA + B) \) has degree \( k \leq n \) and the uniqueness of the solution is obvious. \( \square \)

In the following we shall therefore assume that the matrix pencil \((A, B)\) is regular. Problems of the form (3.2) can be solved using the Weierstraß-Kronecker canonical form.

**Theorem 3.3** Suppose the matrix pencil \((A, B)\) is regular. Then there exist nonsingular matrices \(P\) and \(Q\) such that

\[
P A Q = \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}, \quad P B Q = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix},
\]

where \(N = \text{diag}(N_1, \ldots, N_k)\). Each matrix \(N_i\) is a Jordan block of the form

\[
N_i := \begin{bmatrix} 0 & 1 & & \\ & 0 & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix}, \quad \text{of dimension } m_i.
\]

and \(C\) can be found to be in Jordan canonical form.

This theorem is proved by Gantmacher (cf. [30]). Historically, the work of Gantmacher (cf. [30]) has been an inspiration for the use of matrix pencils in studying DAEs. This work has induced the concept of the so-called index of DAEs, being the most important concept in classifying DAE systems. The notion of index of nilpotency of a matrix pencil can be defined as follows.

**Definition 3.4** The matrix pencil \((A, B)\) has index of nilpotency \( n_i = m \), where \( m = \max_{1 \leq i \leq k} m_i \), i.e. \( m \) is the smallest integer for which \( N^m = 0 \). Then the DAE (3.2) has index of nilpotency \( n_i \).

In the special case that the matrix \(A\) is nonsingular, the DAE (3.2) has index of nilpotency zero. Let the matrix pencil \((A, B)\) be regular. The DAE (3.2) can then be solved, using the Weierstraß-Kronecker canonical form of Theorem 3.3, as follows: premultiply (3.2) by \(P\) and define the coordinate change

\[
x = : Q \begin{bmatrix} u \\ v \end{bmatrix}.
\]

and the transformation

\[
P_g := \begin{bmatrix} r \\ s \end{bmatrix}.
\]

System (3.2) can be written in decoupled form as

\[
\begin{align*}
\dot{u}(t) + Cu(t) = & \quad r(t), \quad (3.6a) \\
Nv(t) + v(t) = & \quad s(t), \quad (3.6b)
\end{align*}
\]
with initial value \([x_0] = \begin{bmatrix} x_0 \\
v_0 \end{bmatrix} = Q^{-1}x_0\). Equation (3.6a) is an ODE. For any initial value \(x_0\) and any continuous function \(r\), it has a unique solution. However, equation (3.6b) is not an ODE. Suppose that the DAE (3.2) has index of nilpotency \(m > 0\) (such that \(N^{m-1} \neq 0\) and \(N^m = 0\)), then we find from (3.6b) by repeated differentiation

\[
v(t) = s(t) - NV(t) \\
= s(t) - Ns(t) + N^2v(t) \\
\vdots \\
= s(t) - Ns(t) + \cdots + (-1)^{m-1}N^{m-1}s^{(m-1)}(t) + (-1)^mN^mv^{(m)}(t),
\]

assuming that \(s\) is sufficiently differentiable. Since \(N^m = 0\), the solution \(v(t)\) of (3.6b) can be written as

\[
v(t) = \sum_{i=0}^{m-1} (-1)^iN^is^{(i)}(t). \tag{3.7}
\]

Thereby, the initial value \(v_0\) must satisfy

\[
v_0 = \sum_{i=0}^{m-1} (-1)^iN^is^{(i)}(0). \tag{3.8}
\]

This means that equation (3.6b) with initial value \(v(0) = v_0\) has a solution only if the initial values are consistent, i.e. if \(v_0\) satisfies (3.8). From the derivation process of the solution \(v\) according to (3.7) of equation (3.6b) it is obvious that not all components of \(v\) necessarily are differentiable. There are some properties, in which DAEs behave differently as compared with ODEs, viz.:

- the initial value \(x_0\) has to be consistent;
- the solution can involve derivatives up to order \(m - 1\) of the forcing function \(g\) if the DAE is of higher index (i.e. \(m \geq 2\));
- the solution \(x\) may consist of components that are not differentiable;
- higher index DAEs may have hidden algebraic constraints.

The latter point can be illustrated by the following example.

**Example 3.5** Consider the linear constant coefficient DAE

\[
\begin{align*}
x_2(t) + x_1(t) &= g_1(t), \tag{3.9a} \\
x_1(t) + x_2(t) &= g_2(t), \tag{3.9b} \\
x_1(t) &= g_3(t). \tag{3.9c}
\end{align*}
\]
3.1. Theory of Differential Algebraic Equations

which consists of one explicit algebraic constraint, namely equation (3.9c). This DAE has the solution

\begin{align*}
x_1(t) &= g_1(t) - g_2(t) + g_3(t). \quad (3.10a) \\
x_2(t) &= g_2(t) - g_3(t). \quad (3.10b) \\
x_3(t) &= g_3(t). \quad (3.10c)
\end{align*}

Apparently there are two hidden algebraic constraints, viz. (3.10a) and (3.10b). □

We proceed with the following definition that will be useful in forthcoming chapters.

**Definition 3.6** Consider the DAE (3.1). Assume that an arbitrary solution component \( x_i \) of the solution \( x \) depends on derivatives of the forcing function \( g \) up to order \( k - 1 \). The component \( x_k \) then is called a variable of index \( k \).

For linear DAEs of the form (3.2), where \( A \) and \( B \) are time dependent, the local index of the pencil \( [A, B] \) is defined as the index of nilpotency of this pencil at any time \( t \). However, this local index does not necessarily determine the structure of these DAE problems as in the case of linear DAEs with constant coefficients. Nevertheless, this local index will play an important role in solving DAEs numerically (as we shall see). For DAEs of the form (3.2) Campbell (cf. [14]) derived an explicit expression for the solution \( x \) in terms of the Drazin inverse of \( A \) and \( B \). This expression does not give further insights than what is already obtained by the results above. The case of rectangular matrices \( A \) and \( B \) has also been studied by Campbell (see [14]).

### 3.1.3 Nonlinear Systems

Next, we consider general DAEs of the form (3.1). The definition of the index of nilpotency (cf. Definition 3.4) has to be extended for these systems. This can be done in various ways. The first and probably most important extension is the so-called differential index (first defined by Gear (cf. [32])).

**Definition 3.7** Consider the general DAE \( f(t, x, \dot{x}) = 0 \). This DAE has differential index \( d_i = m \) if \( m \) is the smallest number of differentiations such that the system of equations

\[
\begin{align*}
f(t, x(t), \dot{x}(t)) &= 0, \\
\frac{df(t, x(t), \dot{x}(t))}{dt} &= 0, \\
&\vdots \\
\frac{d^m f(t, x(t), \dot{x}(t))}{dt^m} &= 0,
\end{align*}
\]

(3.11)

can be transformed into an explicit ODE \( \dot{x}(t) = g(t, x(t)) \) by algebraic manipulations. (This explicit ODE is called the underlying ODE (UODE).)
It is obvious that an ODE has index zero. The index is a measure of the degree of singularity in the system. In general, the higher the index, the more complex the problem and the more difficulties we are likely to encounter in solving the DAE by a numerical method (as we shall see). For linear DAEs with constant coefficients the index of nilpotency and the differential index are equal, i.e. \( ni = di \). For this, consider system (3.6b). After \( m - 1 \) differentiations the solution \( v \) of (3.6b) is obtained. So, after one more differentiation the UODE for \( v \) is obtained. This means that \( di = m = ni \). Furthermore, the initial value \( x(0) = x_0 \) is consistent for equation (3.1) if system (3.11) in the separate variables \( \dot{x}, \dot{x}^2, \ldots, \dot{x}^{m-1} \) has a solution \( \dot{x}(0, x_0) \). System (3.11) shows that every differentiation of the original DAE reduces the index of the new system by one. So, the equation \( \frac{dx}{dt} = 0 \) has index \( m - n \).

Hairer, Lubich and Roche (cf. [41]) have introduced the so-called perturbation index as a measure of the sensitivity of the solutions with respect to perturbations of a given equation (3.1).

**Definition 3.8** Equation (3.1) has a perturbation index \( pi = m, m > 0 \), along a solution \( x \) on \([0, T]\), if \( m \) is the smallest integer such that for all functions \( y \) having a defect

\[
f(t, y(t), \dot{y}(t)) = \delta(t),
\]

the difference between \( x \) and \( y \) on \([0, T]\) is bounded by an estimate of the form

\[
|\|x(t) - y(t)\| - L| |x(0) - y(0)|| + |\|\delta(t)\|| + \ldots + |\|\delta^{m-1}(t)\|| \leq K (|\|x(0) - y(0)|| + \max_{0 \leq \tau \leq T} |\|\delta(t)\|| + \ldots + \max_{0 \leq \tau \leq T} |\|\delta^{m-1}(t)\||),
\]

whenever the expression on the right-hand side is sufficiently small. The constant \( K \) depends only on \( T \) and the length of the interval. Further we say that equation (3.1) has perturbation index zero, if

\[
|\|x(t) - y(t)\| - L| |x(0) - y(0)|| + \max_{0 \leq \tau \leq T} \left| \int_{0}^{\tau} \delta(r) dr \right| \leq K \max_{0 \leq \tau \leq T} \left| \int_{0}^{\tau} \delta(r) dr \right|.
\]

For system (3.6b), the solution \( v(t) \) depends on the \((m - 1)\)th derivative of \( s(t) \). Therefore, for a linear constant coefficient DAE the perturbation index and the index of nilpotency are equal, i.e. \( pi = ni \).

Gear (cf. [33]) has shown that for general DAEs the following relation holds between the perturbation index and the differential index,

\[
di \leq pi \leq di + 1,
\]

for problems (3.1) for which both the differential index and the perturbation index exist. For DAEs which are quasi linear in \( x \) and whose components containing \( x \) are total differentials, i.e.

\[
f(t, x, \dot{x}) = a(t, x)\dot{x} + b(t, x) = 0,
\]

Gear (cf. [33]) has shown that the perturbation index and the differential index coincide, i.e. \( pi = di \).
3.1.4 Semi-Explicit Systems

There are several special classes of implicit DAEs with a relatively simple structure. These subcases can easily be recognized and they often appear in applications. DAEs of the form

\[
\begin{align*}
\dot{x}(t) &= f(t, x(t), y(t)), \\
0 &= g(t, x(t), y(t)),
\end{align*}
\]

where \( x : [0, T] \rightarrow \mathbb{R}^n \) and \( y : [0, T] \rightarrow \mathbb{R}^m \) are called semi-explicit. All other DAEs of the form (3.1) are called fully-implicit DAEs. Semi-explicit DAEs illustrate that DAEs can be considered as systems of differential equations combined with algebraic equations. These algebraic equations define a manifold to which the solution is constrained. Therefore, DAEs can be interpreted as differential equations on manifolds (cf. [71]). For semi-explicit DAEs, the variables can be divided into differential variables and algebraic variables. In the equation above for example, \( x \) is the differential variable and \( y \) is the algebraic variable.

The simplest form of a nonlinear DAE is a semi-explicit index one DAE of the form

\[
\begin{align*}
\dot{x} &= f(x, y), \\
0 &= g(x, y),
\end{align*}
\]

(3.12a)

(3.12b)

with \( \frac{\partial g}{\partial y} \) nonsingular. The solution of this system lies on the manifold defined by (3.12b). Differentiation of the algebraic equation (3.12b) gives

\[
0 = \frac{\partial g}{\partial x} \dot{x} + \frac{\partial g}{\partial y} y.
\]

(3.13)

After substitution of (3.12a) for \( \dot{x} \), equation (3.13) yields

\[
y = -\left(\frac{\partial g}{\partial y}\right) \frac{\partial g}{\partial x} f,
\]

(3.14)

because the Jacobian \( \frac{\partial g}{\partial y} \) is nonsingular. Equations (3.12a) and (3.14) combined constitute the UODE (cf. Subsection 3.1.3) for \( x \) and \( y \). By Definition 3.7, the differential index \( \mu \) is one in this case. Consistent initial values must satisfy \( 0 = g(x_0, y_0) \). Hairer, Lubich and Roche (cf. [41]) have shown that DAE (3.12) has perturbation index \( p \) one.

Consider the semi-explicit system

\[
\begin{align*}
\dot{x} &= f(x, y), \\
0 &= g(x),
\end{align*}
\]

(3.15a)

(3.15b)

where \( \frac{\partial g}{\partial y} \) is nonsingular. Subsequently, differentiating equation (3.15b) and substituting for \( \dot{x} \) results in

\[
0 = \frac{\partial g}{\partial x} f.
\]

(3.16)
Similarly to the DAE (3.12) we see that the DAE (3.15a), (3.16) has differential index one if the matrix \( \frac{\partial f}{\partial x} \) is nonsingular; hence equation (3.15) has differential index two. The solution of DAE (3.15) is not only constrained to lie on the manifold (3.15b), but also on the manifold (3.16). Initial conditions are consistent if they satisfy both equation (3.15b) and equation (3.16). Again, for this system the perturbation index coincides with the differential index as shown in [41].

Under the assumption that \( \frac{\partial f}{\partial x} \) is nonsingular, the problem

\[
\begin{align*}
\dot{x} &= f(x, y), \\
\dot{y} &= k(x, y, z), \\
0 &= g(x),
\end{align*}
\]

where \( x : [0, T] \to \mathbb{R}^n, \ y : [0, T] \to \mathbb{R}^m \) and \( z : [0, T] \to \mathbb{R}^p \), has differential index (and perturbation index) three. Differentiation of equation (3.17c) gives

\[
0 = \frac{\partial f}{\partial x} \dot{x}.
\]

Compare system (3.17a), (3.17b), (3.18) with DAE (3.15). System (3.17a), (3.17b), (3.18) is an index two system, so DAE (3.17) has index three.

In general, the differential index is the most important and most often used definition of the index of a DAE. Therefore, in the following the index will always mean the differential index, and will be denoted by \( \nu \).

### 3.1.5 Applications

DAEs arise in many applications. In Chapter 2 we have shown that they occur in connection with the dynamic analysis of mechanical systems. They also arise in the study of nonlinear circuits and in the study of optimal control problems. Moreover, DAEs are important in investigating the structure of the solutions of singular perturbation problems. In this subsection some of these applications are briefly described.

#### Multibody Systems

The motion of a system of rigid bodies can be described using concepts of classical mechanics. Let \( q \in \mathbb{R}^n \) be a vector of generalized coordinates and \( v \in \mathbb{R}^n \) be a vector of generalized velocities. Assume that the rigid bodies are connected by \( n \) holonomic constraints, which can be expressed as \( \Phi(q, t) = 0 \). The equations of motion for this multibody system can be written as

\[
\begin{align*}
\dot{q} &= v, \\
Mv &= g - \Phi^T \lambda, \\
0 &= \Phi,
\end{align*}
\]

where the matrix \( M \in \mathbb{R}^{n \times n} \) denotes the positive definite mass matrix and the vector \( g \in \mathbb{R}^n \) the applied forces. The unknown Lagrange multipliers \( \lambda \in \mathbb{R}^n \) account for the
3.1. Theory of Differential Algebraic Equations

unknown constraint forces. Moreover, assume that the Jacobian matrix $\phi_q = \frac{\partial \Phi}{\partial q}$ has constant rank $m$. This system of equations constitutes a DAE of index three, as can easily be seen. Indeed, differentiating the constraint (3.19c) twice with respect to time one obtains the velocity constraint

$$\dot{\phi}_q \ddot{q} = -\phi_k := \nu,$$  \hfill (3.20)

and the acceleration constraint

$$\dot{\phi}_q \ddot{q} = -\phi_q \dddot{q} q - 2\phi_q \ddot{q} - \phi_q := y.$$  \hfill (3.21)

Combining equations (3.19b) and (3.21) results in

$$\begin{bmatrix} M & \phi_q^T \\ \phi_q & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} g \\ y \end{bmatrix}.$$  \hfill (3.22)

This system has a unique solution for the accelerations $\ddot{q}$ and the Lagrange multipliers $\lambda$, viz.

$$\ddot{q} = M^{-1} \begin{bmatrix} g - \phi_q^T \lambda \end{bmatrix},$$  \hfill (3.23)

$$\lambda = \left( \phi_q M^{-1} \phi_q^T \right)^{-1} \phi_q M^{-1} g - y.$$  \hfill (3.24)

Substitution of the expression (3.24) for $\lambda$ into (3.23) yields the UODE. Hence,

- the DAE (3.19) with position constraint has index three,
- the DAE (3.19a), (3.19b), (3.20) with velocity constraint has index two,
- the DAE (3.19a), (3.19b), (3.21) with acceleration constraint has index one,

and the equations of motion can be formulated as a DAE of index three, index two, or index one accordingly. All these formulations are mathematically equivalent if the initial values are consistent. The initial values $q_0$, $v_0$ and $\lambda_0$ are consistent if the position constraint (3.19c) and the velocity constraint (3.20) are satisfied and if $\lambda_0$ is determined by (3.24).

Singular Perturbations

There is a close relationship between singular perturbations and DAEs. Consider for example the problem

$$\begin{align*}
\dot{x} &= f(x, y), \\
\varepsilon \dot{y} &= g(x, y), \quad (\varepsilon \neq 0)
\end{align*}$$  \hfill (3.25a)

where $x : [0, T] \to \mathbb{R}^n$ and $y(t) : [0, T] \to \mathbb{R}^m$ and where the forcing functions $f$ and $g$ are sufficiently smooth vector functions of the same dimensions as $x$ and $y$, respectively. Setting $\varepsilon = 0$ in (3.25a) one obtains the reduced DAE

$$\begin{align*}
\dot{x} &= f(x, y), \\
0 &= g(x, y).
\end{align*}$$  \hfill (3.26a)
Suppose that the Jacobian $\frac{\partial}{\partial y}$ is invertible in a neighbourhood of the solution. Then the DAE (3.26) has index one (as explained in Subsection 3.1.4). By the Implicit Function Theorem, equation (3.26b) possesses a locally unique solution $y = G(x)$, say. Substitution of this solution into equation (3.26a) results in the ODE

$$\dot{x} = \Gamma(x, G(x)).$$

(3.27)

This is the so-called state space form (SSF). It is well known that the solution of system (3.25a) possesses a power series expansion in $\varepsilon$, cf. [61], with smooth $\varepsilon$ independent coefficients $x_0$ and $y_0$

$$x(t) = x_0(t) + \varepsilon x_1(t) + \varepsilon^2 x_2(t) + \ldots + \varepsilon^N x_N(t) + O(\varepsilon^{N+1}),$$

(3.28a)

$$y(t) = y_0(t) + \varepsilon y_1(t) + \varepsilon^2 y_2(t) + \ldots + \varepsilon^N y_N(t) + O(\varepsilon^{N+1}).$$

(3.28b)

where $O(\varepsilon^n)$ is to be understood as bounded in norm. Substitution of equations (3.28a) and (3.28b) into system (3.25a) and comparing the powers of $\varepsilon$ leads to the following system of equations

$$x_0 = f(x_0, y_0),$$

(3.29a)

$$0 = g(x_0, y_0).$$

$$\dot{x}_1 = f_1(x_0, y_0)x_1 + f_2(x_0, y_0)y_1,$$

(3.29b)

$$\dot{y}_0 = g_1(x_0, y_0)x_1 + g_2(x_0, y_0)y_1,$$

$$\vdots$$

$$\dot{x}_N = f_N(x_0, y_0)x_1 + f_N(x_0, y_0)y_1 + \psi(x_0, y_0, \ldots, x_{N-1}, y_{N-1}),$$

(3.29m)

$$\dot{y}_{N-1} = g_N(x_0, y_0)x_1 + g_N(x_0, y_0)y_1 + \psi(x_0, y_0, \ldots, x_{N-1}, y_{N-1}).$$

System (3.29a) is an index one DAE, system (3.29a), (3.29b) is a DAE of index two, and DAE (3.29) has index $l + 1$. Hence, the study of singular perturbations gives rise to DAEs of arbitrarily high index.

### 3.2 Multistep Methods

Consider the ODE

$$\dot{x} = f(t, x), \quad t > 0,$$

(3.30a)

subject to the initial condition

$$x(0) = x_0.$$

(3.30b)
3.2. Multistep Methods

A general $k$-step multistep method to solve IVP (3.30) numerically is

$$\sum_{i=0}^{k} c_i x_n - \sum_{i=0}^{k} \beta_i f(t_{n-i}, x_{n-i}),$$

where $h$ denotes the step-size, i.e. $t_n = nh$ and where $c_i$, $\beta_i$, $i = 0, \ldots, k$ are known constants. If $\beta_0 = 0$ then the scheme is explicit; otherwise it is implicit. Using the so-called shift operator $E$, i.e. $Ex = x_{n+1}$, we can write the above relation in a more compact form

$$\rho(E)x_{n+k} = h\sigma(E)f(t_{n-k}, x_{n-k}),$$

where the so-called generating polynomials $\rho$ and $\sigma$ of the multistep method are defined by

$$\rho(z) := \sum_{i=0}^{k} c_i z^{-i} \quad \text{and} \quad \sigma(z) := \sum_{i=0}^{k} \beta_i z^{-i}. \quad (3.31)$$

For $\beta_i = 0$, $i = 1, \ldots, k$, $\beta_0 = 1$ one finds the $k$-step backward difference formulae (BDF), where $c_i$, $i = 0, \ldots, k$ are the coefficients of the BDF method (these coefficients can be found in any textbook on the numerical solution of ODEs, see e.g. [42]). For ODEs the $k$-step BDF method is of order $k$ and it is stable for $k < 7$.

Consider the index one DAE

$$Ax + Bx = g, \quad t \geq 0. \quad (3.32)$$

Application of Euler forward to (3.32) yields the difference equation

$$A(x_n - x_{n-1}) + hBx_{n-1} = hg_{n-1}.$$

Since $A$ is singular this equation is not solvable. However, applying the implicit Euler method to the problem above we obtain

$$A(x_n - x_{n-1}) + hBx_n = hg_n. \quad (3.33)$$

Assume that the matrix pencil $(A, B)$ is nonsingular, i.e. $\lambda A + B$ is regular except for a finite number of $\lambda \neq 0$. Then, there exists a $h_0 > 0$ such that the equation (3.33) is solvable for all step-sizes $h$, $0 < h < h_0$. The method above can easily be extended to implicit methods like the $k$-step BDF method, i.e.

$$A\rho(E)x_{n+k} + hBx_n = hg_n. \quad (3.34)$$

3.2.1 Constant Coefficient DAEs

One can easily prove the following theorem (cf. [91]) for linear constant coefficients DAEs.
Theorem 3.9 The k-step BDF method (k ≥ 6) with constant step-size applied to linear DAEs with constant coefficients (cf. (3.2)) of index ν (ν ≥ 1) is convergent of order νhν after (ν − 1)k + 1 steps.

So, the numerical solution converges in an interval bounded away from the initial time. This convergence result is not easily extendable to variable step-sizes, because the error estimates used in BDF codes are not realistic for DAEs of higher index; moreover the solution is not accurate at the first two steps after a change in the step-size as can be seen from the following example. In particular, the backward Euler method fails to converge at the end of the first step following a change in the step-size, as can be seen from the following example (cf. [9]).

Example 3.10 Consider the index three problem
\[ \begin{align*}
    x_1 &= x_2, \\
    x_2 &= x_1, \\
    0 &= x_1 - f.
\end{align*} \]

Applying Euler backward results in
\[ \begin{align*}
    x_{1,n} &= f(t_n), \\
    x_{2,n} &= \frac{1}{h_n} (f(t_n) - f(t_{n-1})), \\
    x_{3,n} &= \frac{1}{h_n} \left( \frac{f(t_n) - f(t_{n-1})}{h_n} - \frac{f(t_{n-1}) - f(t_{n-2})}{h_{n-1}} \right),
\end{align*} \]

where \( h_n = t_n - t_{n-1} \). Notice that \( x_{3,n} \) is a wrong approximation of \( f(t_{n-1}) \), because of the division by \( h_n \) instead of \( \frac{1}{2}(h_n + h_{n-1}) \). This results in an error given by
\[ \frac{1}{2} \left( \frac{h_{n-1}}{h_n} - 1 \right) f(t_{n-1}). \]

This means that \( x_{3,n} \) converges to \( f(t_n) \) with accuracy \( O(h_n^2) \) if the step-size is effectively constant, i.e. \( h_n = h_{n-1}(1 + O(h_{n-1})) \). However, if \( h_n = O(h_{n-1}) \) with \( h_n \neq h_{n-1} \) then this results in an error \( O(1) \) and Euler backward does not converge.

However, Gear et al. (cf. [35]) have shown for variable step-size BDF methods that if the ratio of subsequent steps is kept bounded, then the global error in the numerical solution for the k-step BDF method applied to linear DAEs with constant coefficients of index \( ν \) is \( O(h_{\text{max}}^q) \), where \( q = \min(k, \nu + 2) \). Nevertheless, Euler backward will fail integrating even a simple linear constant coefficient DAE of index three, as we would then find a global error of \( O(1) \).

The foregoing can be generalized to general problems of the form (3.1). Applying the k-step BDF method leads to the following difference equation
\[ \frac{1}{h_n} \rho(E)x_{n-1} = 0. \]

This equation is solvable if the matrix pencil \( \left( \begin{array}{c} \frac{d}{dt} \end{array}, \frac{d}{dt} \end{array} \right) \) at \( t_n \) is regular. In this case, the local index at \( t_n \), denoted as \( \nu_n \), is the index of the local pencil \( \left( \begin{array}{c} \frac{d}{dt} \end{array}, \frac{d}{dt} \end{array} \) at \( t_n \).
### 3.2.2 Index One Systems

Due to the close relationship between singular perturbed ODEs and DAEs (cf. Subsection 3.1.5), a multistep method for semi-explicit DAEs of index one can be obtained from applying a multistep method to the singular perturbation (3.25a) and afterwards letting \( \epsilon \) approach zero, i.e. the so-called direct approach (cf. [43]). This results in the difference equations

\[
\sum_{i=0}^{k} \omega_i x_{n-i} = h \sum_{i=0}^{k} \beta_i f(x_{n-i}, y_{n-i}) \tag{3.36a}
\]

\[
0 = h \sum_{i=0}^{k} \beta_i g(x_{n-i}, y_{n-i}) \tag{3.36b}
\]

and the following theorem (cf. [43]) holds.

**Theorem 3.11** Suppose that system (3.12), has a nonsingular Jacobian \( \frac{\partial f}{\partial \bar{x}} \). Consider a multistep method of order \( k \) which is stable at the origin and at infinity and suppose that the error of the starting values \( x_0 \) and \( y_0 \), for \( j = 0, 1, \ldots, k - 1 \) is \( O(h^k) \). Then the global error of (3.36), satisfies

\[
x_n - x_t = O(h^k) \quad \text{and} \quad y_n - y_t = O(h^k),
\]

for \( t_n = nh \leq \text{Const.} \).

In this direct approach, \( x_n \) and \( y_n \) will usually not satisfy the constraint \( g(x, y) = 0 \), because equation (3.36b) is not necessarily equivalent to this. Therefore, in the so-called indirect approach (3.36b) is replaced by

\[
0 = g(x_n, y_n). \tag{3.37}
\]

It follows from the Implicit Function Theorem that Theorem 3.11 holds for the indirect approach and the condition that infinity is in the stability region of the multistep method can be dropped. Therefore, even explicit methods can be applied.

Another important class of index one systems is the class of uniform index one systems, i.e. fully-implicit index one systems (3.1) with constant rank \( \frac{\partial f}{\partial \bar{x}} \) and whose index is identically equal to one in a neighborhood of the solution. Gear and Petzold (cf. [37]) have proved the following result for these uniform index one DAEs.

**Theorem 3.12** Let (3.1) be a uniform index one DAE, and assume that \( f \) is differentiable with respect to \( x \) and \( \bar{x} \). Then the solution of (3.1) by the \( k \)-step BDF method with fixed step-size for \( k \leq 6 \) converges to \( O(h^k) \) if all initial values are correct of order \( O(h^k) \).

One can show (cf. [34]) that if variable step-size BDF methods are implemented in such a way that the method is stable for ODEs, then the \( k \)-step BDF method \( (k \leq 6) \) is convergent for fully-implicit index one DAEs. Gressier and März (cf. [40]) have studied
the application of general linear multistep and one-leg methods to index one DAEs. They obtained convergence results for various formulations of the multistep methods applied to fully implicit index one DAEs. The BDF codes DASSL (cf. [66]) and LSODI (cf. [48]) have been developed to solve DAEs of index zero and index one numerically.

3.2.3 Semi-Explicit Index Two Systems

For higher index systems it is impossible to obtain convergence in general, even for backward Euler. Therefore, attention has been focused on higher index systems of a special structure, such as semi-explicit index two DAEs and index three systems of Hessenberg form (cf. Subsection 3.2.4). In this subsection the behaviour of multistep methods applied to semi-explicit index two problems is studied. Let

\[ \begin{align*}
    x &= f(x, y), \\
    0 &= g(x),
\end{align*} \tag{3.38a} \tag{3.38b} \]

where \( f \) and \( g \) are assumed to be sufficiently differentiable, and where \( \frac{\partial y}{\partial x} \) is assumed to be invertible in a neighbourhood of the solution. Hence, the DAE (3.38) has index two. Again, a linear multistep method system can be applied in two different ways to the above DAE, viz.

\[ \begin{align*}
    \sum_{i=0}^{k} a_i x_{n-i} &= h \sum_{i=0}^{k} \beta_i f(x_{n-i}, y_{n-i}), \\
    0 &= g(x_n),
\end{align*} \tag{3.39a} \tag{3.39b} \]

or replacing (3.39b) by

\[ 0 = \sum_{i=0}^{k} \beta_i g(x_{n-i}). \tag{3.40} \]

This last equation is the analogue of equation (3.36b).

Hairer and Wanner (cf. [43]) have shown that a multistep method (3.39) of order \( p \) applied to DAE (3.38), yields a local error \( \mathcal{O}(h^{p+1}) \) for the \( x \) component and a local error \( \mathcal{O}(h^p) \) for the \( y \) component. For BDF methods the study of convergence is simpler than for general multistep methods because the \( x \) - and \( y \) - component can be treated separately (due to \( \beta_0 = \ldots = \beta_{k-1} = 0 \)). The following convergence result (cf. [9]) holds for BDF methods.

**Theorem 3.13** Suppose the nonlinear semi-explicit index two DAE (3.38), is to be solved by the \( k \)-step BDF method (\( k \leq 6 \)). Then the \( k \)-step BDF method is convergent of order \( k \), i.e. \( x_n - x(t_n) = \mathcal{O}(h^k) \), \( y_n - y(t_n) = \mathcal{O}(h^k) \), after \( k + 1 \) steps, whenever the initial values satisfy \( x_i - x(t_i) = \mathcal{O}(h^{k+1}) \), for \( i = 0, \ldots, k - 1 \).
If variable step-size BDF methods are implemented in such a way that they are stable for ODEs, then the k-step BDF method \( k \leq 6 \) converges for semi-explicit index two DAEs (cf. [34]). The main result for general multistep methods of the form (3.39), is the following (cf. [43]).

**Theorem 3.14** Consider the index two system (3.38), where \( \frac{\phi}{m} \) is assumed to be invertible in a neighbourhood of the solution. Assume that the k-step multistep method is stable and strictly stable at infinity. If the k-step multistep method has order \( p \geq 2 \), then the global error satisfies

\[
x_n - x(t_n) = O(h^p), \quad y_n - y(t_n) = O(h^p),
\]

whenever the initial values satisfy

\[
x_i - x(t_i) = O(h^{k+1}), \quad \text{for} \quad i = 0, \ldots, k - 1.
\]

The previous results can easily be extended to the second approach, with equation (3.39b) replaced by (3.40).

### 3.2.4 Index Three Systems of Hessenberg Form

In previous subsections it was noted that BDF methods converge for fully implicit index one systems, semi-explicit index two systems and for linear constant coefficient DAEs of arbitrary index with the same accuracy as for standard ODEs. In the following convergence results for Hessenberg systems of size three are discussed.

In general, the DAE (3.1) is in **Hessenberg form of size m** if it can be written as

\[
\begin{align*}
x_1 &= f_1(x_1, x_2, \ldots, x_m, t), \\
x_2 &= f_2(x_2, x_3, \ldots, x_{m-1}, t), \\
\vdots & \\
x_i &= f_i(x_{i-1}, x_i, \ldots, x_{m-1}, t), \\
\vdots & \\
0 &= f_m(x_{m-1}, t),
\end{align*}
\]

where the matrix \( \frac{\partial f_1}{\partial x_1} \frac{\partial f_1}{\partial x_2} \cdots \frac{\partial f_1}{\partial x_m} \) is nonsingular in a neighbourhood of the solution. Assume that the \( f_i \) are sufficiently smooth. Then the above Hessenberg system of size \( m \) is a DAE of index \( m \) and is solvable. Apparently, Hessenberg index three systems can be expressed as

\[
\begin{align*}
x_1 &= f_1(x_1, x_2, x_3, t), \\
x_2 &= f_2(x_1, x_2, t), \\
0 &= f_3(x_2, t),
\end{align*}
\]
where the matrix \( \mathbf{A}, \mathbf{B}, \mathbf{C} \) is invertible in a neighbourhood of the solution. Multibody systems (see (3.19)) are Hessenberg index three systems. Brenner and Engquist (cf. [10]) have shown that BDF methods have \( k \)-th order convergence for sufficiently accurate initial values.

**Theorem 3.15** Suppose the Hessenberg index three system (3.43) is solved by a \( k \)-step BDF method \( k \leq 6 \) with constant step-size. Let the initial values be consistent of order \( k + 1 \), i.e.

\[
\begin{align*}
x_{i+1} - x_i \ &= \ O(h^{k+1}), \\
x_{2i} - x_i \ &= \ O(h^{k+1}), \\
J_i \ &= \ O(h^{k+1})
\end{align*}
\]

for \( i = 0, 1, \ldots, k - 1 \), and the algebraic equations be solved with accuracy \( O(h^{k+1}) \) for \( k \geq 2 \), and \( O(h^{k+3}) \) for \( k = 1 \). Then the \( k \)-step BDF method is convergent of order \( k + 1 \) after \( k + 1 \) steps.

For semi-explicit index two systems the convergence results could be extended to variable step-sizes (cf. the previous subsection). However, for Hessenberg index three systems the convergence result stated above cannot be extended to hold for variable step-sizes, because a new boundary layer of reduced convergence rates is initiated each time the step-size is changed. In particular, the backward Euler method fails to converge at the end of the first step following a change in the step-size, as can be seen from Example 3.10.

In the foregoing we have shown that BDF methods with constant step-size applied to several classes of DAE systems have \( k \)-th order of convergence, provided the initial values are consistent and the functions are sufficiently smooth. BDF codes with variable step-size can be used to solve these DAEs. However, difficulties occur, since discretization of DAEs of higher index results in ill-conditioned iteration matrices (see Section 6.2). By scaling the variables and the equations this problem can be remedied somewhat. The convergence test and error test must also be modified to allow a BDF code to solve this type of problems (cf. Section 6.3). For a detailed discussion we refer to [11] and [12].

### 3.3 Runge-Kutta Methods

In any textbook on the numerical solution of ODEs, see e.g. [13, 42], one can find results for Runge-Kutta methods applied to ODEs. In this thesis we have confined ourselves to multistep and BDF methods. For completeness we give a literature overview (a more detailed overview is given in [82]) on the results of implicit Runge-Kutta methods (IRKs) applied to DAEs. Petzold (cf. [67]) was the first person to study the convergence of IRK methods applied to linear constant coefficient DAEs. For IRK methods applied to semi-explicit index one DAEs (3.12) the orders of convergence of the \( x \)-component are the same as for ODEs. However, in general the \( y \)-component does not attain the same order.
of convergence as for ODEs. In the indirect approach (cf. also (3.37)) IRK methods obtain the same order of convergence as for ODEs, since they satisfy the constraints on each step. In the direct approach (see also (3.36)) the algebraic variable $y$ can generally suffer from order reduction (see [9, 40, 41, 43]). In this approach stiffly accurate methods are favourable, since they obtain the same order of convergence as for ODEs (cf. [18, 40]), because they have the property that the constraints are satisfied exactly at the end of each time step. For fully implicit index one DAEs there is an extra loss of accuracy which can occur due to mixing between the errors in the differential and the algebraic parts of the DAE system. Therefore, IRK methods are, in general, less accurate for fully implicit index one DAEs than for semi-explicit index one DAEs (cf. [9]). In these results the strict stability condition (cf. [40, 41]) is very important. Symmetric methods such as the implicit midpoint rule, which do not satisfy the strict stability condition, can be unstable for this class of problems (cf. [40]).

The first convergence results for RK-methods applied to index two DAEs (3.15) have been obtained by Petzold (cf. [67]). Replacing $y$ in (3.15a) by $z$ transforms the index two system (3.15) in an index one system. Because of this close relationship between semi-explicit index two systems and uniform index one systems it follows that the global error in the differential variable $x$ the conclusions above hold. For the algebraic variable $y$ order results are given in [67]; these results have been improved in [41]. Hairer and Wanner (cf. [43]) have given a complete set of order results based on a nontrivial extension of Butcher’s theory of rooted trees. In [43] order results are given for the case of a singular RK-matrix, for Rosenbrock methods and for extrapolation methods. The order of convergence for the $y$ component is generally quite poor for these methods (cf. [41]). A more serious problem is that these methods can suffer from oscillations and instabilities (see [2]). This problem of instability, oscillation and order reduction can be solved by applying so-called Projected Implicit Runge-Kutta methods (PIRKs) (cf. [3]) to semi-explicit index two systems. These PIRKs project the numerical solution on the constraint. For stiffly accurate RK-methods the projected and unprojected RK-methods coincide. The stability and the order of convergence are recovered by means of this extra projection onto the constraint, and one can prove superconvergence for projected collocation methods.

The convergence of IRKs applied to Hessenberg index three systems (cf. (3.43)) has been studied in [41]. It is shown that these methods can suffer from order reduction and therefore care must be taken in choosing a RK method appropriate for DAEs.
CONDITIONING OF DIFFERENTIAL ALGEBRAIC EQUATIONS OF INDEX ONE

For DAEs it is very important to study the concept of conditioning, since thinking of the differential index (see Definition 3.7) one performs as many differentiations till some (Jacobian) matrix is nonsingular. This mathematical concept may be not very useful in numerical practice, in particular when this matrix becomes almost singular. We may therefore, provisionally, conclude that also the constant $\kappa$ as appearing in the definition of the perturbation index (cf. Definition 3.8), may be an inappropriate stability constant.

In this chapter we more precisely study the conditioning of DAEs of index one, i.e. the influence of perturbations on the solution. After considering index one linear DAEs in a general setting, semi-explicit systems are considered. For these problems stability constants are derived. Those constants appear to depend on the norm of the inverse of a Jacobian matrix. This implies that the stability constant grows unboundedly when this matrix becomes almost singular. Under appropriate circumstances, such a DAE is close to a DAE of index higher than one. When this higher index problem is well-conditioned its sensitivity constants are not large. Intuitively, therefore, it seems not probable that the stability constant of the index one system grows unboundedly. Therefore, a more realistic conditioning concept is developed in the case that the aforementioned Jacobian matrix is almost singular, showing that this DAE effectively behaves like an associated higher index system, including its stability behavior.

### 4.1 GENERAL LINEAR DAEs OF INDEX ONE

In this chapter we study the conditioning of initial value problems (IVPs) for DAEs of index one. Griepentrog and März have investigated this stability question for the general linear case in [40] by rewriting the DAE as an equivalent system of state equations, which is a useful general start for our analysis. So, consider the general linear DAE

\[ \dot{\mathbf{y}} + \mathbf{B}\mathbf{y} = \mathbf{q}, \]  

(4.1a)
subject to the initial value
\[ \mathbf{x}(0) = \mathbf{x}_0, \quad (4.1b) \]

where the solution \( \mathbf{x} : [0, T] \to \mathbb{R}^n \), the coefficients \( \mathbf{A}, \mathbf{B} : [0, T] \to \mathbb{R}^{n \times m} \) are continuous matrix functions and the forcing function \( \mathbf{q} : [0, T] \to \mathbb{R}^m \) is a continuous vector function. Let the matrix \( \mathbf{A} \) be singular on the interval \( I := [0, T] \) and have a smooth null space; i.e., there exists a continuously differentiable projector \( \mathbf{Q}(t) \) onto \( N(t) := \text{null}(\mathbf{A}(t)) \). Let the projector \( \mathbf{P}(t) \) be defined by \( \mathbf{P} := \mathbf{I}_n - \mathbf{Q} \). It is well-known (cf., e.g., [40]) that the DAE \((4.1a)\) has global index one if and only if the matrix
\[ \mathbf{G}(t) := \mathbf{A}(t) + \mathbf{B}(t)\mathbf{Q}(t), \quad (4.2) \]
is nonsingular for all \( t \in I \). In this case, the IVP \((4.1)\) is uniquely solvable for arbitrary forcing functions \( \mathbf{q} \in C \) and initial values \( \mathbf{x}(0) \in \text{range}(\mathbf{P}(0)) \). Note that \( \mathbf{A}\mathbf{Q} = \mathbf{0} \), and therefore, solutions of such an index one DAE belong to the function space
\[ \mathcal{C} := \{ \mathbf{x} \in C | \mathbf{P}\mathbf{x} \in C^1 \}. \quad (4.3) \]

Let \( \mathcal{S}(t) := \{ \mathbf{x} \in \mathbb{R}^n | \mathbf{P}(t)\mathbf{x} \in \text{range}(\mathbf{A}(t)) \} \), and let \( \mathbf{Q}_0(t) := \mathbf{G}^{-1}(t)\mathbf{B}(t) \) i.e. the so-called canonical projector onto \( N(t) \) along \( \mathcal{S}(t) \). The so-called state variable is defined by \( \mathbf{u} := \mathbf{P}\mathbf{x} \). Denote \( \mathbf{P}_0(t) := \mathbf{I}_n - \mathbf{Q}_0(t) \). The variable \( \mathbf{x} \) can be written in terms of \( \mathbf{u} \) as
\[ \mathbf{x} = \mathbf{P}_0\mathbf{u} + \mathbf{Q}_0\mathbf{G}^{-1}\mathbf{q}, \quad (4.4) \]
Using the aforementioned definitions, the IVP \((4.1)\) can be rewritten as the following equivalent state equation
\[ \dot{\mathbf{u}} = (\mathbf{P}_0\mathbf{A} - \mathbf{P}_0\mathbf{G}^{-1}\mathbf{B})\mathbf{u} + (\mathbf{P}_0 + \mathbf{P}_0\mathbf{Q}_0\mathbf{G}^{-1})\mathbf{q}, \quad (4.5a) \]
subject to the initial condition
\[ \mathbf{Q}_0\mathbf{u}(0) = \mathbf{0}. \quad (4.5b) \]

Clearly, the stability of the index one IVP \((4.1)\) is governed by the state equation \((4.5a)\), and in particular by the matrix function \( \mathbf{P}_0\mathbf{A} - \mathbf{P}_0\mathbf{G}^{-1}\mathbf{B} \).

Stability of linear IVPs for DAEs of index one can now be formulated in terms of the growth behaviour of fundamental solutions and kinematic eigenvalues of the regular state space ODE \((4.5a)\). Hence, on a finite interval \( I \) no rapidly increasing modes can exist, and that, on an infinite interval \( I \), fundamental solution modes of \((4.5a)\) cannot become unbounded when \( t \) approaches infinity. The fundamental solution \( \mathbf{X} \), say, is uniquely determined by the IVP
\[ \dot{\mathbf{X}}(t) + \mathbf{B}(t)\mathbf{X}(t) = \mathbf{0}, \quad \mathbf{P}(0)[\mathbf{X}(0) - \mathbf{L}_n] = \mathbf{0}. \quad (4.6) \]

This \( \mathbf{X} \) can be expressed in terms of the fundamental solution \( \mathbf{U} \) of \((4.5a)\), which satisfies
\[ \dot{\mathbf{U}} - (\mathbf{P}_0\mathbf{A} - \mathbf{P}_0\mathbf{G}^{-1}\mathbf{B})\mathbf{U}, \quad \mathbf{U}(0) = \mathbf{L}_n. \]
4.1. General Linear DAES of Index One

It can easily be verified that

$$\text{null}(\bar{X}(t)) = N(0), \quad \text{range}(\bar{X}(t)) = S(t),$$

and

$$X(t) = P_t U(t) \bar{U}(t) P(0).$$

The solution $\bar{X}$ of problem (4.1) is given by

$$\bar{X}(t) = \bar{X}(t) \bar{x}_0 + P_e(t) \int_0^t \bar{U}(t) \bar{U}^{-1}(s) \left[ \bar{P}(s) + \bar{P}(s) \bar{Q}(s) \bar{G}^{-1}(s) \right] \bar{q}(s) ds + \bar{Q}(t) \bar{G}^{-1}(t) \bar{q}(t). \quad (4.7)$$

As in [54] the following quantities can be introduced in order to study the notion of conditioning

$$\bar{k}_1 = \max \{ ||\bar{X}(t)||, t \in I \},$$
$$\bar{k}_2 = \sup \{ \int_0^t ||\bar{U}(t) \bar{U}^{-1}(s)||^2 ds : t \in I \},$$
$$\bar{k}_3 = \max \{ ||\bar{P}(t)||, t \in I \},$$
$$\bar{k}_4 = \max \{ ||\bar{Q}(t)||, t \in I \},$$
$$\bar{k}_5 = \max \{ ||\bar{G}(t)||, t \in I \},$$
$$\gamma_5 = \max \{ ||P(t)||, t \in I \},$$

(4.8)

The conditioning constants $\bar{k}_1$ and $\bar{k}_2$ are already known from the standard theory of ODEs. $\bar{k}_3$ and $\bar{k}_4$ measure the growth of fundamental solutions, i.e., they are related to the stability of the system. The extra quantities $\bar{k}_3, \bar{k}_4, \bar{k}_5$ and $\gamma_5$ deal with the specific character of DAES (as compared to regular ODEs) and measure in a way how much this DAE deviates from an ODE. By (4.7) the solution can be bounded in terms of the inhomogeneities; thus

$$||\bar{X}||_\infty \leq \bar{k}_1 ||\bar{x}_0|| + \gamma_5 \bar{k}_2 ||\bar{q}||_2 + \bar{k}_4 ||\bar{q}||_\infty,$$

(4.9)

where $\frac{1}{p} + \frac{1}{q} = 1$. We have arrived at the following definition.

**Definition 4.1** The initial value problem (4.1) is called well-conditioned if the conditioning constant $\kappa := \max(\bar{k}_1, \gamma_5 \bar{k}_2 ||\bar{q}||_2 + \bar{k}_4 ||\bar{q}||_\infty)$ is of moderate size.

From equation (4.5a) it is obvious that $\bar{G}^{-1}$ plays a crucial role in determining whether the IVP (4.1) is well-conditioned or not. Consider the case of an almost singular $\bar{G}$, i.e., $\bar{G}^{-1}$ is large in norm, say $||\bar{G}^{-1}|| = \epsilon^{-1}$, $|\epsilon| \rightarrow 0$. In this case we expect the state equation (4.5a) to have large Lipschitz constants and so, small perturbations of $\bar{q}$ may seemingly lead to large perturbations of the solution $\bar{X}$. This general setting is not easily accessible for more detailed analysis, therefore we shall restrict ourselves to the semi-explicit case in order to obtain more insight.
4.2 SEMI-EXPLICIT INDEX ONE DAEs

Semi-explicit linear DAEs are not only easier than the general case but also still quite meaningful, as we only encounter semi-explicit forms e.g. in multibody dynamics. Hence, we study DAEs of the form

\[
x(t) = A(t)x(t) + B(t)y(t) + p(t), \\
0 = C(t)x(t) + D(t)y(t) + q(t),
\]

with \( x \in \mathbb{R}^n, y \in \mathbb{R}^m, \) and where the matrix \( D \) is nonsingular for each \( t \in I \). Hence, the DAE (4.10a) has index one. It is clear that one can only prescribe an initial condition \( x(0) \), say, for the dynamic variable \( x \) at initial time

\[
x(0) = x_0.
\]

In this case, the aforementioned matrices are given by

\[
\hat{Q} = \text{diag}(O, I_n), \quad \hat{G} = \begin{bmatrix} I_n & -B \\ O & -D \end{bmatrix}, \quad \hat{G}^{-1} = \begin{bmatrix} I_n & -BD^{-1} \\ O & -D^{-1} \end{bmatrix},
\]

\[
\hat{P} \hat{G}^{-1} = \begin{bmatrix} I_n & -BD^{-1} \\ O & O \end{bmatrix}, \quad \hat{P} \hat{G}^{-1} \hat{B} = \text{diag}(BD^{-1}C - A, O),
\]

\[
\hat{Q} = \begin{bmatrix} O & O \\ D^{-1}C & I_n \end{bmatrix}, \quad \hat{P} = \begin{bmatrix} I_n & O \\ -D^{-1}C & I_n \end{bmatrix}.
\]

System (4.10) yields (with \( \hat{s} = [\hat{x}] \))

\[
x = \hat{s}, \quad y = -D^{-1}Cu - D^{-1}q,
\]

and

\[
u = (A - BD^{-1}C)u + p - BD^{-1}q, \quad \text{with } u(0) = x_0.
\]

Thus, the semi-explicit index one case is equivalent to the following system of equations

\[
\dot{x} = (A - BD^{-1}C)x + p - BD^{-1}q, \quad \text{with } x(0) = x_0, \quad (4.11a)
\]

\[
y = -D^{-1}Cx - D^{-1}q. \quad (4.11b)
\]

Here, the solution can be bounded as follows

\[
\|x\|_{\infty} \leq \kappa_1\|x_0\| + \kappa_2\|p - BD^{-1}q\|_{\infty}, \tag{4.12a}
\]

\[
\|y\|_{\infty} \leq \|D^{-1}C\|\|x\|_{\infty} + \|D^{-1}\|\|q\|_{\infty}
\leq \kappa_1\|D^{-1}C\|\|x_0\| + \kappa_2\|D^{-1}C\|\|p - BD^{-1}q\|_{\infty} + \|D^{-1}\|\|q\|_{\infty}. \tag{4.12b}
\]
4.3. Conditioning for Almost Singular $\mathbf{D}$

where

$$
\kappa_2 = \max \{ \| \mathbf{X}(t) \|, \ t \in I \}, \\
\kappa_2 = \sup \{ \int_0^1 \| \mathbf{X}(t) \mathbf{X}^{-1} (s) \| \, ds, \ t \in I \},
$$

(4.13)

and where $\mathbf{X}(t)$ is the fundamental solution of the state space ODE (4.11a). In the definition of $\kappa_2$ (cf. (4.13)) we have used the 1-norm since this is useful for problems where $\mathbf{X}$ exhibits layer behaviour (cf. the rest of this chapter). Note that these estimates coincide with the estimates in (4.9). Especially,

$$
\dot{\mathbf{X}}(t) = \dot{\mathbf{P}}_t(t) \dot{\mathbf{U}}(t) \mathbf{P}(0) = \begin{bmatrix} \mathbf{I}_n & \mathbf{O} \\ \mathbf{D}^{-1} \mathbf{C} & \mathbf{O} \end{bmatrix} \mathbf{U}(t) := \begin{bmatrix} \mathbf{I}_n & \mathbf{O} \\ \mathbf{D}^{-1} \mathbf{C} & \mathbf{O} \end{bmatrix} \left[ \begin{bmatrix} \mathbf{X}(t) & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_n \end{bmatrix} \right],
$$

which implies that $\mathbf{X}$, and as a consequence $\kappa_1$, correspond to the $x$-components of the general fundamental solution matrix $\dot{\mathbf{X}}$ and the general conditioning constant $\kappa_1$, respectively. Moreover, $\kappa_2$ corresponds to the $x$-components of the Green function $\dot{\mathbf{U}}(t) \mathbf{U}^{-1}(s)$.

For the $y$-components this means that the factor $\| \mathbf{D}^{-1} \mathbf{C} \|$ follows from the corresponding components of $\| \dot{\mathbf{P}}_t \|$, whereas the last term in the estimate (4.12b) for the $y$-component coincides with the second component of $\dot{\mathbf{Q}} \mathbf{G}^{-1} \dot{\mathbf{q}}$. From (4.12) we can conclude that the IVP (4.10) is well-conditioned if the conditioning constants $\kappa_1$ and $\kappa_2$, as well as $\| \mathbf{D}^{-1} \mathbf{C} \|$ and $\| \mathbf{D}^{-1} \|$ are of moderate size. We already pointed out that the conditioning of problem (4.1) is determined by ODE (4.5a) and its fundamental solution.

Remark 4.2 Notice that (4.12) might overestimate the conditioning of equation (4.10) to a large extent when the matrix $\mathbf{D}$ is almost singular. In that case we may need a thorough estimation of the convolution-integral in order to quantify the conditioning of the system. In the next section we proceed with a more careful study of the conditioning in the case that we deal with an almost singular matrix $\mathbf{D}$ in order to develop a realistic conditioning concept for this case.

### 4.3 Conditioning for Almost Singular $\mathbf{D}$

In this section we consider the conditioning of a semi-explicit index one IVP (4.10) with a coefficient matrix $\mathbf{D}$ being almost singular. The growth behaviour of the dynamic variable $\mathbf{x}$ of this system is governed by the state ODE (4.11a). By defining the matrix $\hat{\mathbf{A}}$, which describes the homogeneous part of (4.11a), i.e.

$$
\hat{\mathbf{A}} := \mathbf{A} - \mathbf{BD}^{-1} \mathbf{C},
$$

(4.14)

the stability of the DAE can be characterized by considering this matrix $\hat{\mathbf{A}}$. To simplify the discussion we assume to have $\mathbf{D}$ such that $\mathbf{D} = \varepsilon \tilde{\mathbf{D}}$, where $\| \tilde{\mathbf{D}}^{-1} \| \approx 1$ ($\varepsilon < 0$). The constraint equation of (4.10) can then be written as

$$
\theta = \tilde{\mathbf{D}}^{-1} \mathbf{Cx} + \varepsilon \mathbf{y} + \tilde{\mathbf{D}}^{-1} \mathbf{q}.
$$
Without loss of generality $\tilde{D}$ can be taken to be the identity matrix, and $\tilde{A}$ reads
\begin{equation}
\tilde{A} = A - \varepsilon^{-1} BC.
\end{equation}

Suppose that the matrix $CB$ is nonsingular for all $t$, $0 \leq t \leq T$. All coefficient matrices and the sign of $\varepsilon$ are assumed to be such that the IVP considered is a uniform stable one. Next, let $\tilde{p}$ be defined as $\tilde{p} := p - \varepsilon^{-1} B q$. Using the definitions above, the state ODE (4.11a) can be written as
\begin{equation}
\dot{x} = \tilde{A} x + \tilde{p}.
\end{equation}

with solution
\begin{equation}
x(t) = \Phi(t)x(0) + \int_0^t \Phi(t)\Phi^{-1}(\tau)\tilde{p}(\tau) d\tau,
\end{equation}
where $\Phi$ denotes the fundamental solution of (4.16). For $y$ one finds
\begin{equation}
y = -\varepsilon^{-1} C \Phi(t)x(0) - \varepsilon^{-1} C \int_0^t \Phi(t)\Phi^{-1}(\tau)\tilde{p}(\tau) d\tau - \varepsilon^{-1} q(t).
\end{equation}

For illustrative reasons we first consider the constant coefficient scalar case here, i.e. the following index one DAE
\begin{equation}
\begin{align*}
\dot{x} &= ax + by + p, \\
0 &= cx + dy + q,
\end{align*}
\end{equation}
where $d \neq 0$. For this simple system the governing state ODE reads
\begin{equation}
\begin{align*}
\dot{x} &= (a - bd^{-1}c)x + p - bd^{-1}q, \\
\dot{y} &= \tilde{a} \dot{x} + \tilde{p},
\end{align*}
\end{equation}
where $\tilde{a}$ and $\tilde{p}$ are defined similarly to $\tilde{A}$ and $\tilde{p}$, respectively. The corresponding solution reads
\begin{equation}
x(t) = e^{\tilde{A} t} x(0) + \int_0^t e^{\tilde{A}(t-\tau)} \tilde{p}(\tau) d\tau.
\end{equation}

Let $d = \varepsilon$, $|\varepsilon| \downarrow 0$, and note that
\begin{equation}
\begin{align*}
\int_0^t e^{\tilde{A}(t-\tau)} \tilde{p}(\tau) d\tau &= -\tilde{a}^{-1} (be^{-1}q(t) - e^{\tilde{A} \varepsilon^{-1}} q(0)) \\
&\quad + \tilde{a}^{-1} \int_0^t e^{\tilde{A}(t-\tau)} be^{-1} q(\tau) d\tau.
\end{align*}
\end{equation}

Assume that $|cb|$ is bounded away from zero. Using (4.22) in (4.21), we obtain to first order in $\varepsilon$
\begin{equation}
x(t) = e^{\tilde{A} t} x(0) + \int_0^t e^{\tilde{A}(t-\tau)} p(\tau) d\tau \\
- \frac{1}{\varepsilon} \left[ 1 + \frac{\tilde{a}}{be} + O(\varepsilon^2) \right] (q(t) - e^{\tilde{A} \varepsilon^{-1}} q(0)) - \int_0^t e^{\tilde{A}(t-\tau)} q(\tau) d\tau.
\end{equation}
4.3. CONDITIONING FOR ALMOST SINGULAR D

The growth behaviour of $x$ is therefore characterized by the following estimate
\[ |x(t)| \leq \kappa_1 |x(0)| + \varepsilon \kappa_2 \max_{t \leq r \leq t} |p(t)| + \frac{1}{|c|} \left( 1 + \frac{a}{bc} \right) \varepsilon + O(\varepsilon^2) \left( \max_{t \leq r \leq t} |q(t)| + \kappa_1 |q(0)| + \varepsilon \kappa_2 \max_{t \leq r \leq t} |q(t)| \right) \]
(4.24)

where the constant $\kappa_1$ is defined like in (4.13), i.e.
\[ \kappa_1 := \max_{t \leq r \leq t} \frac{1}{a - bc^{-1}e^{-t}}. \]
(4.25)

The conditioning constant $\kappa_2$ (cf. (4.13)) can be seen to be of order of magnitude $\varepsilon$. For that reason a more useful quantity is $\hat{e}_2$, defined by
\[ \hat{e}_2 := d^{-1} \max_{t \leq r \leq t} \int_0^t e^{a - bc^{-1}e^{-s}} ds \, ds \, dr. \]
(4.26)

It is easily seen that $\hat{e}_2$ is an $O(1)$ constant as $d = e \downarrow 0$. Note that for $d$ bounded away from zero, $\hat{e}_2$ and $e_2$ are quite similar again as $d \hat{e}_2 = \kappa_2$; the solution $x$ (cf. (4.21)) then can be estimated as
\[ |x(t)| \leq \kappa_1 |x(0)| + \kappa_2 \max_{t \leq r \leq t} |p(t)| + \kappa_2 |b| d^{-1} \max_{t \leq r \leq t} |q(t)|. \]
(4.27)

**Remark 4.3** The estimate (4.24) shows that $x$ exhibits initial layer behaviour for $d = e$, ($e \downarrow 0$). Furthermore, it shows that $x$ depends on $q$ and on $q$ times a factor of order $e$.

From equation (4.19) we find that $y$ satisfies
\[ y = -d^{-1} cx - d^{-1}q. \]
(4.28)

Substitution of (4.21) gives
\[ y = -d^{-1} cx(0) - d^{-1} \int_0^t e^{a - bc^{-1}e^{-s}} p(s) ds - d^{-1} q(t). \]
(4.29)

For $d = e$, ($e \downarrow 0$), this yields
\[ y(t) = -e^{-1} c e^{a - bc^{-1}e^{-t}} x(0) - e^{-1} q(t) - e^{-1} \int_0^t e^{a - bc^{-1}e^{-s}} p(s) ds - \frac{e^{-1} c}{a - be^{-1} c} (be^{-1} q(t) - e^{a - bc^{-1}e^{-t}} e^{-1} q(0)) + \frac{e^{-1} c}{a - be^{-1} c} \int_0^t e^{a - bc^{-1}e^{-s}} b e^{-1} q(s) ds. \]
\[ y(t) = e^{a - bc^{-1}e^{-t}} \left( y(0) - q(0) \left( \frac{a}{bc} + O(e) \right) \right) - e^{-1} c \int_0^t e^{a - bc^{-1}e^{-s}} p(s) ds + \left( \frac{a}{bc} + O(e) \right) q(t) - e^{-1} \left( 1 + \frac{a}{bc} + O(\varepsilon^2) \right) \int_0^t e^{a - bc^{-1}e^{-s}} q(s) ds. \]
(4.31)
So, the algebraic variable \( y \) can be bounded as follows
\[
|y(t)| \leq \kappa_1 \left( |y(0)| + \max_{\theta \leq \tau \leq T} |p(\tau)| \right) + \kappa_2 \max_{\theta \leq \tau \leq T} |q(\tau)| \\
+ \left( \frac{a}{\beta c} + O(\varepsilon) \right) \max_{\theta \leq \tau \leq T} |q(\tau)| + \left( \frac{a}{\beta c} + O(\varepsilon^2) \right) \kappa_2 \max_{\theta \leq \tau \leq T} |q(\tau)|
\]  
(4.32)

For the algebraic variable \( y \), equation (4.29) shows that the following estimate holds when \( d = O(1) \)
\[
|y(t)| \leq \kappa_1 \frac{d^{-1} c}{d^{-1} c} ||x(0)|| + \kappa_2 \frac{d^{-1} c}{d^{-1} c} \max_{\theta \leq \tau \leq T} |p(\tau)|| + \kappa_3 \frac{d^{-1} c}{d^{-1} c} \max_{\theta \leq \tau \leq T} |q(\tau)| \\
+ \frac{d^{-1} c}{d^{-1} c} \max_{\theta \leq \tau \leq T} |q(\tau)|.
\]  
(4.33)

**Remark 4.4** The estimate (4.32) shows that \( y \) exhibits initial layer behaviour in case that \( d = O(1) \) and for \( \delta b^{-1} \) not large. Moreover, it shows that \( y \) effectively depends on \( q \) then. Of course this is what one should expect, since these conditions are almost requiring that the problem is index two.

In the remainder of this section the general linear case is considered.

In the sequel we will need the following

**Lemma 4.5** Consider the homogeneous ODE
\[
\dot{y} = Ay.
\]  
(4.34)

and a slightly perturbed one
\[
\dot{z} = \Delta z.
\]  
(4.35)

where
\[
\Delta = A + \delta A.
\]

Let \( \Phi, \Phi \) denote the fundamental solutions of (4.34), (4.35), respectively. Assume that \( \Phi(0) = \Phi(0) = I \). Let \( \delta A \leq \varepsilon A \) and \( \delta_2 \varepsilon A < 1 \). Then
\[
\delta_1 \leq \frac{\kappa_1}{1 - \kappa_2 \varepsilon A},
\]  
(4.36)

where \( \kappa_1 \) and \( \varepsilon \) denote the conditioning constants for the fundamental solutions \( \Phi \) and \( \Phi \), respectively.

**Proof.** A homogeneous solution of (4.35) satisfies
\[
z(t) = \Phi(t)z(0) + \int_0^t \Phi(t)\Phi^{-1}(s)\delta A(s)z(s) ds.
\]  
(4.37)
4.3. Conditioning for Almost Singular D

As a consequence,

$$\Phi(t) = \Phi(r) + \int_0^t \Phi(r) \Phi^{-1}(s) \delta A(s) \Phi(s) ds.$$  \hspace{1cm} (4.38)

This yields inequality (4.36).

In the general linear case we prove the following theorem.

**Theorem 4.6** Consider the IVP (4.10) given by the index one DAE (4.10a) subject to initial condition (4.10b). The conditioning of this problem is given by the estimates (4.12). Let $D = \varepsilon I$ ($\varepsilon \downarrow 0$) and let $CB$ be nonsingular with all eigenvalues bounded away from zero. Assume that the DAE is a uniform stable one. Let the matrix coefficients $A$, $B$ and $C$ vary slowly. Then the conditioning of this IVP is given by the estimates

$$\|x\| \leq \kappa_1 \|x(0)\| + \kappa_2 \|p\| + \|B(CB)^{-1}\| \left(\|q\| + \delta_1 \|q(0)\|\right)$$

$$+ \varepsilon \delta_2 \|B(CB)^{-1}\| \|q\| \quad (\varepsilon \downarrow 0),$$  \hspace{1cm} (4.39)

and

$$\|y\| \leq \kappa_1 \|y(0)\| + \kappa_2 \|p\| + \delta_2 \|q\| \quad (\varepsilon \downarrow 0),$$  \hspace{1cm} (4.40)

respectively. The conditioning constants are defined by

$$\kappa_1 := \max \left\{ \| \exp(-\varepsilon^{-1}BCt) \|, \quad 0 \leq t \leq T \right\},$$

$$\kappa_2 := \max \left\{ \| \exp(-\varepsilon^{-1}CBt) \|, \quad 0 \leq t \leq T \right\},$$  \hspace{1cm} (4.41)

and

$$\delta_2 := \sup \left\{ \int_0^T \| \exp(-\varepsilon^{-1}BC(t-r)) \| \, dr, \quad 0 \leq t \leq T \right\},$$

$$\delta_1 := \varepsilon^{-1} \sup \left\{ \int_0^T \| \exp(-\varepsilon^{-1}CB(t-r)) \| \, dr, \quad 0 \leq t \leq T \right\}.$$  \hspace{1cm} (4.42)

respectively.

**Proof.** The state ODE (4.11) can be written as

$$\varepsilon x = -(B(r)C(r) - \varepsilon A(r)x - B(r)q(r)) + \varepsilon p(r).$$  \hspace{1cm} (4.43)

The matrix coefficients are assumed to vary slowly (in particular bounds for derivatives are small compared to $\varepsilon$) and highly oscillatory solutions are excluded from the system. This implies that the dominant solution behaviour is determined by fast inner solutions in an initial layer, while the solution is smooth in the outer region. Since the matrix coefficients are varying slowly, the general solution behaviour of $x$ is effectively determined by the homogeneous matrix coefficient case

$$\varepsilon A(0) = -B(0)C(0) + \varepsilon A(0).$$  \hspace{1cm} (4.44)
Hence, in the sequel the constant coefficient case is considered.

Lemma (4.5) shows that if we would take the matrix $A$ equal to the zero-matrix $O$, only higher order terms would be neglected. Hence, in order to simplify the analysis further, the matrix $A$ is assumed to be zero. As a consequence, in the remainder of this proof we consider the following constant coefficient DAE

\[
\begin{align*}
\dot{x} &= By + p, \\
0 &= Cx + \varepsilon y + q.
\end{align*}
\]  

(4.45)

As shown before, only higher order effects are neglected this way. For equation (4.45) this implies that $\hat{A} = -\varepsilon^{-1}BC$, and the stability behaviour of (4.10) is governed by $\hat{A}$, at least in the initial layer. By defining the matrix $T$ and its inverse as

\[
T := \begin{bmatrix} B & B^L \end{bmatrix},
\]  

(4.46)

($B^L$ denoting a full rank matrix with columns orthonormal to those of $B$), and

\[
T^{-1} = \begin{bmatrix} (B^T B)^{-1}B^T \\ (B^L)^T \end{bmatrix},
\]  

(4.47)

respectively, the rank-deficient matrix $BC$ can be factorized as

\[
BC = T \begin{bmatrix} CB & CB^L \\ 0 & 0 \end{bmatrix} T^{-1}.
\]  

(4.48)

Since the matrix coefficients are slowly varying, the fundamental solution matrix $\Phi$ can be written as

\[
\Phi(t) = \exp(-\varepsilon^{-1}BCt)\Phi(0).
\]  

(4.49)

Without loss of generality $\Phi(0)$ can be taken equal to the identity matrix. The fundamental solution matrix $\exp(-\varepsilon^{-1}BCt)$ reads

\[
\exp(-\varepsilon^{-1}BCt) = T \begin{bmatrix} \exp(-\varepsilon^{-1}CBt) & (CB)^{-1}(\exp(-\varepsilon^{-1}CBt) - I)CB^L \\ 0 & 1 \end{bmatrix} T^{-1}.
\]  

(4.50)

The integral

\[
J := \varepsilon^{-1} \int_0^t \exp(-\varepsilon^{-1}BC(t - \tau))Bq(\tau) d\tau
\]  

(4.51)

can be worked out as follows

\[
J = \varepsilon^{-1}TS[t - \tau]T^{-1}Bq(\tau) \bigg|_0^t - \varepsilon^{-1}\int_0^t S[t - \tau]T^{-1}Bq(\tau) d\tau,
\]  

(4.52)
where $S$ is defined as

$$
S(t) := 
\begin{bmatrix}
  \varepsilon(CB)^{-1}e^{(-\varepsilon^{-1}CB) t} & (CB)^{-1}(\varepsilon(CB)^{-1}e^{(-\varepsilon^{-1}CB) t} + I)CB^{-1} \\
  0 & -I
\end{bmatrix}.
$$

(4.53)

From $T^{-1}B = [I]$, it follows that

$$
J = B(CB)^{-1}(q(t) - e^{-\varepsilon^{-1}CB}q(0)) - B(CB)^{-1}\int_0^t e^{-\varepsilon^{-1}CB(t-\tau)}q(\tau) d\tau.
$$

(4.54)

Therefore, the state variable $x$ satisfies

$$
\begin{align*}
x(t) & = \exp(-\varepsilon^{-1}BC)x(0) + \int_0^t \exp(-\varepsilon^{-1}BC(t-\tau))p(\tau) d\tau \\
& \quad - B(CB)^{-1}(q(t) - \exp(-\varepsilon^{-1}CB)q(0)) \\
& \quad + B(CB)^{-1}\int_0^t \exp(-\varepsilon^{-1}CB(t-\tau))q(\tau) d\tau.
\end{align*}
$$

(4.55)

So $x$ can be bounded by

$$
\|x\| \leq \kappa_1\|x(0)\| + \kappa_2\|p\| + \|B(CB)^{-1}\|\|q\| + \varepsilon\|B(CB)^{-1}\|\|q\|,
$$

(4.56)

where the conditioning constants are defined by (4.41) and (4.42).

Similarly, one finds for the algebraic variable $y$ the expression

$$
y(t) = -\varepsilon^{-1}C\exp(-\varepsilon^{-1}BC)x(0) - \varepsilon^{-1}C\int_0^t \exp(-\varepsilon^{-1}BC(t-\tau))p(\tau) d\tau \\
- \varepsilon^{-1}\exp(-\varepsilon^{-1}CB)q(0) - \varepsilon^{-1}\int_0^t \exp(-\varepsilon^{-1}CB(t-\tau))q(\tau) d\tau.
$$

(4.57)

Since

$$C\exp(-\varepsilon^{-1}BCt) = \exp(-\varepsilon^{-1}CB)tC,$$

this results in

$$
y(t) = \exp(-\varepsilon^{-1}CB)y(0) - \varepsilon^{-1}\int_0^t \exp(-\varepsilon^{-1}CB(t-\tau))Cp(\tau) d\tau \\
- \varepsilon^{-1}\int_0^t \exp(-\varepsilon^{-1}CB(t-\tau))q(\tau) d\tau
$$

(4.58)

This implies that $y$ can be bounded as

$$
\|y\| \leq \varepsilon_1\|y(0)\| + \varepsilon_2\|C\|\|p\| + \varepsilon_2\|q\|.
$$

(4.59)
Remark 4.7 Taking $A = 0$ implies that $\|y\|$ does not depend on $\|q\|$. However, letting $A$ be a nontrivial matrix would mean that $\|y\|$ depends on $\|q\|$, but it will make the proof of the theorem a lot more technical.

Remark 4.8 From the stability assumptions in Theorem 4.6 it follows that the eigenvalue of $CB$ are moderately small, i.e. bounded away from zero. This implies that both $k_1$ and $k_1$ are equal to 1, and that the conditioning constant $\kappa_2$ is a conditioning constant of $O(1)$ for $0 < \varepsilon \ll 1$.

Remark 4.9 Both $x$ and $y$ may exhibit initial layer behaviour, when $D = \varepsilon I$, $(\varepsilon \perp 0)$. Then $y$ depends upon the first derivative of $q$, whereas $x$ depends on $q$ and on $\dot{q}$ times a factor of $O(\varepsilon)$.

Remark 4.10 For $0 < \varepsilon \ll 1$ the matrix $CB$, which is important in the conditioning of index two DAEs (cf. Chapter 5), appears into the conditioning of DAEs of index one.

From (4.12) it seemed that the stability constants would grow unboundedly as $\varepsilon^{-1}$ $(\varepsilon \perp 0)$. However, we have shown in Theorem 4.6 that the sensitivity constants are in fact moderately small (and not dominated by $\varepsilon^{-1}$) when the DAE is \textit{close to a DAE of index two}, i.e. $CB$ is regular, such that the DAE effectively is of index two. This can be generalized to the case that the matrix pencil $(A, B)$ (see Subsection 3.1.2) is regular for $\varepsilon \perp 0$, since regularity of the pencil for $\varepsilon \perp 0$ means that the DAE is close to a DAE of higher index. This again implies that the stability constants are in fact of order one and are not dominated by $\varepsilon^{-1}$. However, when the matrix pencil $(A, B)$ becomes singular when $\varepsilon \perp 0$, the stability constants might be dominated by $\varepsilon^{-1}$ as can be seen from a simple example.

Example 4.11 Consider the IVP 4.10 where

\[
A = \begin{bmatrix}
-1 & -1 \\
0 & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
1 \\
0
\end{bmatrix}, \quad C = \begin{bmatrix}
0 & 1
\end{bmatrix} \quad \text{and} \quad D = \varepsilon.
\]

(4.60)

Note that $CA/B = 0 \forall t \in N$. Moreover, $\det(\lambda A - B) = -\varepsilon(\lambda + 1)^2$, so the matrix pencil $(\bar{A}, \bar{B})$ is singular for $\varepsilon = 0$. The solution of this index one DAE reads

\[
x(t) = \begin{bmatrix}
-\varepsilon^{-1} & (1 + \varepsilon^{-1})(e^{-t} - 1) \\
0 & 1
\end{bmatrix} x(0) - \varepsilon^{-1} \int_0^t \begin{bmatrix}
e^{-s} - 1 \\
0
\end{bmatrix} q(s) ds
+ \int_0^t \begin{bmatrix}
e^{-t-s} - 1 \\
1
\end{bmatrix} p(s) ds,
\]

\[
y(t) = -\varepsilon^{-1} x_2(0) - \varepsilon^{-1} \int_0^t p(s) ds - \varepsilon^{-1} q(t).
\]

It is obvious that the stability constants are of order $O(\varepsilon^{-1})$ for $x$ as well as for $y$. \hfill \Box

See Example 6.9 for a mechanical example of such a system.

In the previous analysis, the conditioning of an index one DAE (4.10) with an almost singular matrix $D$, i.e. $D = \varepsilon I$ $(\varepsilon \perp 0)$ was studied into more detail. The dynamic behaviour of the system is governed by the matrix $\bar{A}$, cf. (4.15). This might suggest that
the dynamic behaviour of the DAE is very sensitive to perturbations of the matrix coefficients, i.e. A and B. However, this is disproved by the following

**Theorem 4.12** Consider the index one DAE (4.10a). Let \( D = \varepsilon I, \varepsilon \downarrow 0 \) and let \( CB \) be nonsingular. Let the matrix coefficients A, B and C vary slowly. Assume that the problem has no highly oscillatory solutions. Assume that this DAE is a stable problem. Then the stability behaviour of this DAE will only be altered slightly by small perturbations of A, B as well as C.

*Proof.* The stability behaviour of the system is governed by the matrix \( \tilde{A} \), cf. (4.15). Hence, the stability is effectively determined by \(-\varepsilon^{-1}BC\). The matrices \( CB \) and \( BC \) have the same nonzero eigenvalues. Small perturbations of \( B \) and \( C \) yield small perturbations of both \( BC \) and \( CB \). As a result, only the nontrivial eigenvalues are perturbed slightly. Since the system was assumed to be stable, all eigenvalues of \( CB \) are moderately small and bounded away from zero. This implies that the eigenvalues of the perturbed system still lie in the stable region and that the dynamic behaviour changes only to a minor amount. \( \square \)

Hence, for index one DAEs which are nearly index two and which are well conditioned in the latter sense, perturbations of the coefficient matrices will be controlled by the same stability constants. In Chapter 5 we will show that the situation is completely different for index two DAEs which are nearly index three. From the proof of Theorem 4.12 we conclude that the effect of slight perturbations of the coefficients B and C may be large when \(|CB| = O(\varepsilon)|\), implying that the index one DAE is close to a DAE of index higher than two. For index one DAEs with a matrix pencil \((\tilde{A}, B)\) which becomes singular when \( \varepsilon \downarrow 0 \) a simple example shows that slight perturbations of A, B or C can have a dramatic effect on the dynamic behaviour of the DAE.

**Example 4.13** Consider the same DAE as in Example 4.11. Let the matrix \( B \) be perturbed into \( B = [I] \). The trivial eigenvalue of \( A - BD^{-1}C \) is then perturbed DAE into \(-\delta/e\) having a dramatic impact on stability when \( \delta/e < 0 \). \( \square \)

### 4.4 ASYMPTOTIC ANALYSIS OF ALMOST HIGHER INDEX DAEs

In this section we will show that the solution of the semi-explicit index one system (4.10) with an almost singular matrix \( D \) can be close to the solution of an associated higher index system. This explains why the conditioning of such a DAE depends on \( q \) and its derivatives. Let us focus our attention to index one DAEs being close to DAEs of index two. Using a singular perturbation approach one can show that, after an initial layer, the solution of such a DAE approaches the solution of the associated index two system. As a consequence, such an index one DAE is called close to an index two system. As in
Section 4.3 we consider the DAE

\[ \dot{x} = Ax + By + p, \]
\[ 0 = Cx + \varepsilon \hat{D}y + q, \]  

(4.61)

subject to the initial condition \( x[0] = x_0 \). Let us assume that \( CB \) is invertible for all \( t \in [0, T] \). Naturally, all coefficient matrices and the sign of \( \varepsilon \) are supposed to be such that the latter IVP is stable. After differentiating the constraint of system (4.61), substitution of the differential equation yields the equivalent system

\[ \dot{x} = Ax + By + p, \]
\[ \varepsilon \hat{D}y = -(C + CA)x - (CB + \varepsilon \hat{D})y - q - Cp. \]  

(4.62)

This singularly perturbed ODE can be analysed by a singular perturbation approach using asymptotic power series expansions as in [62]. The system (4.62) has an \((m + \eta)\)-dimensional solution space, parameterized by any prescribed bounded initial vector \( x(0) \).

On fixed bounded intervals \( 0 \leq t \leq T \), the solution has the asymptotic form

\[ x(t) = a(t) + \alpha(t, \varepsilon), \]
\[ y(t) = b(t) + \varepsilon^{-1} \beta(t, \varepsilon), \]  

(4.63)

where \( \tau = t/\varepsilon \) is the so-called stretched variable. Here, the solution is decoupled into the slow solution vectors \( a \) and \( b \) and the potentially fast solution vectors \( \alpha \) and \( \beta \). The outer solution \( [\mathbf{a}(\tau), \mathbf{b}(\tau)] \) has an asymptotic power series expansion

\[ \sum_{j=0}^{\infty} \left[ \begin{array}{c} a_j(t) \\ b_j(t) \end{array} \right] \varepsilon^j, \]  

(4.64)

which formally satisfies the system (4.62), whereas the initial layer correction \( \left[ \begin{array}{c} \alpha(t) \\ \varepsilon^{-1} \beta(t) \end{array} \right] \) has an asymptotic expansion

\[ \sum_{j=0}^{\infty} \left[ \begin{array}{c} \alpha_j(t) \\ \varepsilon^{-1} \beta_j(t) \end{array} \right] \varepsilon^j, \]  

(4.65)

whose terms all decay exponentially to zero as the stretched variable \( \tau = t/\varepsilon \) tends to infinity. After substitution of the inner and outer solution into system (4.62) one can show that the fast inner solution \( \tau \) satisfies the homogeneous system

\[ \frac{d\alpha}{d\tau} = \varepsilon A\alpha + B\beta, \]
\[ \frac{d\beta}{d\tau} = -\varepsilon \hat{D}^{-1}(C + CA)\alpha - \hat{D}^{-1}(CB + \varepsilon \hat{D})\beta. \]  

(4.66)
on \( \tau \geq 0 \), subject to the initial condition
\[
\beta(0, \varepsilon) = -\tilde{D}^{-1}(0) C(0) x(0) - \tilde{D}^{-1}(0) q(0) - \varepsilon b(0, \varepsilon). \tag{4.67}
\]

Then, the leading term \([\mathbf{a}_0]^{\infty}\) of the initial layer correction satisfies
\[
\begin{align*}
\frac{d\mathbf{a}_0}{d\tau} &= \mathbf{B}(0) \beta_0, \\
\frac{d\beta_0}{d\tau} &= -\tilde{D}^{-1}(0) C(0) \mathbf{B}(0) \beta_0,
\end{align*}
\tag{4.68a}
\]
subject to the initial condition
\[
\beta_0(0) = -\tilde{D}^{-1}(0) [C(0) x(0) + q(0)]. \tag{4.68b}
\]

This is trivial if and only if \( x(0) \) lies on the constraint of the associated index two constraint \( Cx + q = 0 \). This yields for \( \beta_0(\tau) \)
\[
\beta_0(\tau) = \exp[-\tilde{D}^{-1}(0) C(0) \mathbf{B}(0) \tau] \beta_0(0) \tag{4.69a}
\]
while \( \mathbf{a}_0(\tau) \) reads
\[
\mathbf{a}_0(\tau) = -\mathbf{B}(0) [C(0) \mathbf{B}(0)]^{-1} \tilde{D}(0) \beta_0(\tau). \tag{4.69b}
\]

Hence,
\[
\mathbf{a}_0(0) = \mathbf{B}(0) [C(0) \mathbf{B}(0)]^{-1} [C(0) x(0) + q(0)]. \tag{4.70}
\]

Both limiting inner solutions (4.69) are exponentially decaying to zero as the stretched variable \( \tau \) tends to infinity. Relation (4.70) determines \( \mathbf{a}_0(0) = x(0) - \mathbf{a}_0(0) \) which is required to specify the limiting outer solution \( \mathbf{a}_0(t) \) completely. The limiting outer solution \([\mathbf{a}(0)]\) is uniquely determined by the differential system
\[
\dot{\mathbf{a}}_0 = A \mathbf{a}_0 + \mathbf{B} \mathbf{b}_0 + \mathbf{p}, \tag{4.71a}
\]
the initial value
\[
\mathbf{a}_0(0) = (I - \mathbf{P}) [0] x(0) - \mathbf{B}(0) [C(0) \mathbf{B}(0)]^{-1} q(0), \tag{4.71b}
\]
and the relation
\[
\mathbf{b}_0 = -(C \mathbf{B})^{-1} (C \mathbf{A}) \mathbf{a}_0 + C \mathbf{p} + \dot{q}. \tag{4.71c}
\]

Here \( \mathbf{P} := \mathbf{B}(CB)^{-1} C \). Since the initial value \( \mathbf{a}_0(0) \) satisfies the constraint \( Cx + q = 0 \),

system (4.71a), (4.71c) is equivalent to the following index two system
\[
\begin{align*}
\dot{\mathbf{a}}_0 &= A \mathbf{a}_0 + \mathbf{B} \mathbf{b}_0 + \mathbf{p}, \\
0 &= C \mathbf{a}_0 + q. \tag{4.72}
\end{align*}
\]
In general, one can show that all coefficients $a_i$ and $b_i$ for $i \geq 0$ in the asymptotic series expansion of the outer solution satisfy index two systems similar to the latter one. So, the solution is decoupled into an inner and an outer solution. The slowly varying limiting outer solution is shown to satisfy the reduced index two system (4.72), while in the initial layer the limiting inner solution is exponentially decaying to zero. This implies that after an initial layer of thickness $e$ the solution of (4.61) will approach the solution of the index two system (4.72). However, if the initial value $x(0)$ lies on the constraint of the associated index two system, then it is obvious that the leading terms of the outer solution satisfy the latter index two system, whereas the leading terms of the inner solution vanish. Hence, the index one system (4.61) is close to the reduced index two system

$$\begin{align*}
x &= Ax + By + p, \\
0 &= Cx + q
\end{align*}$$

and as a consequence the index one system will behave effectively like this associated index two system. Consequently, both the limiting outer solution $b_0$ and the algebraic variable $y$ depend on the first derivative of the inhomogeneity $q$. This is completely in agreement with the conclusion of Section 4.3 that the growth behaviour of $y$ depends on the derivative of $q$.

For system (4.61) arbitrary initial values for $x$ should be provided. Then the initial values for $y$ can be determined by the algebraic constraint equation of (4.61). However, the closeness of (4.61) to the index two system (4.73) makes the situation more complex. Since by perturbing (4.61) slightly by setting $e = 0$, the initial values $x_0$ have to be chosen such that $x_0 = x_0 - B(0)(C(0)B(0))^{-1}q(0)$, where $x_0 \in \text{range}(I - P)(0)$, can be chosen arbitrarily. Then the initial values of $y$ are being determined by

$$y_0 = -(C(0)B(0))^{-1}(C(0) + C(0)A(0))x_0 + C(0)p(0) + q(0)$$

obtained from differentiating (4.73b) and substituting (4.73a) into it. If, for $0 < e \ll 1$, arbitrary initial values are prescribed for $x$ then this almost implies inconsistency of the initial values. That means that an initial layer adjustment with very fast changing components $x$ and $y$ is needed to satisfy the constraint equation of system (4.61). Such an initial layer is not at all needed if the initial values are chosen to be consistent. Obviously, this holds also for a more general index one system (4.10) with a matrix $D$ that is almost singular. In general, such an index one DAE is close to a DAE of higher index.

### 4.5 INDEX ONE DAEs close to Higher Index DAEs

In the preceding section we have shown that DAEs of lower index can be close to DAEs of higher index. Effectively after a small initial layer these problems of lower index will behave like the associated system of higher index. For analysing and solving this sort of problems it is important to know the effective index of the algebraic variables. In this
section, a method is derived to determine the effective index of the variables, at least in theory. Consider the DAE system:

\[
\begin{align*}
\dot{x} &= Ax + By + p, \\
\quad \theta &= Cx + Dy + q.
\end{align*}
\] (4.75)

where \( x: [0, T] \to \mathbb{R}^n \), \( y: [0, T] \to \mathbb{R}^m \), and the matrices \( A, B, C \) and \( D \) have suitable dimensions. Assume that \( \|C\| = 1 \) and that \( \|D^{-1}\| = \varepsilon^{-1}, (\varepsilon \downarrow 0) \). From [38] we deduce that there exist matrices \( X, S, T \), and \( V \), such that

\[
\begin{align*}
C &= XSU^T, \\
D &= XT V^T, \\
S &= \text{diag}(s_1, \ldots, s_m), \quad s_i > 0, \\
T &= \text{diag}(t_1, \ldots, t_m), \quad t_i \geq 0.
\end{align*}
\] (4.76)

where \( U \in \mathbb{R}^{n \times n} \) and \( V \in \mathbb{R}^{m \times m} \) are orthogonal matrices, \( X \in \mathbb{R}^{n \times n} \) is an invertible matrix and \( SS^T + TT^T = SS^T + T^2 = I_m \). This is called the generalized singular value decomposition (GSVD).

We may call \( r \) the numerical rank of \( D \) if the following holds:

\[
\begin{align*}
t_1 \geq \cdots \geq t_r \gg \varepsilon \geq t_{r+1} \geq \cdots \geq t_m \geq 0.
\end{align*}
\]

Partition \( T \) and \( V \) according to

\[
T = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix}, \quad \text{with} \quad T_{11} = \text{diag}(t_1, \ldots, t_r), \quad \text{and} \\
T_{22} = \text{diag}(t_{r+1}, \ldots, t_m)
\]

and

\[
V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}, \quad \text{with} \quad V_1 \in \mathbb{R}^{n \times r}, \quad V_2 \in \mathbb{R}^{m \times (m-r)}.
\]

Define \( z = [z_1] \), with \( z_1 = V_1^T y \) and \( z_2 = V_2^T y \), respectively. This shows that the vector \( z_1 \) represents the index one variables, whereas \( z_2 \) contains the variables of effectively higher index (as explained before). Thus, system (4.75) can be written as

\[
\begin{align*}
\dot{x} &= Ax + Bz_1 + BV_2 z_2 + p, \\
\quad \theta &= Cx + XT z_1 + q.
\end{align*}
\] (4.77a)

So, the algebraic constraint (4.77b) can be expressed as

\[
\theta = SU^T x + Tz + \bar{q}.
\] (4.78)

where \( \bar{q} \) is defined as \( \bar{q} := X^{-1} q \). This implies that the DAE (4.77a), (4.78) is equivalent to

\[
\begin{align*}
\dot{x} &= Ax + Bz_1 + BV_2 z_2 + p, \\
\quad \theta &= [SU^T]_1 x + T_1 z_1 + \tilde{q}_1, \\
\quad \theta &= [SU^T]_2 x + T_2 z_2 + \tilde{q}_2.
\end{align*}
\] (4.79a)
Replacing the algebraic equation (4.79c) by the reduced system (i.e. ε = 0)
\[ \theta = (SU^T)x + \theta_3, \] (4.80)
enables us to determine the index of the variable ν. For the matrix \( E := (SU^T)B^0V_2 \) we assume that \( ||(E)^{-1}|| \leq M \) (where \( M \) is a constant of moderate size). This implies that \( z_3 \) contains index two variables. However, if \( E \) is singular or nearly singular, say \( ||(E)^{-1}|| = \varepsilon^{-1} (\varepsilon \neq 0) \), then \( z_3 \) may contain variables of index three or higher. Again, we may perform an SVD, now of \( E \). We can distinguish two cases, viz. in the first case let
\[ \tilde{U}^T \tilde{E} \tilde{V} = \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \text{with} \begin{bmatrix} \Sigma = \text{diag}(\sigma_i), \; i = 1, \ldots, l, \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \text{with} \begin{bmatrix} \sigma_1 \geq \ldots \geq \sigma_l > 0 \end{bmatrix}. \] (4.81)

Introducing the corresponding partitionings
\[ \tilde{U}(SU^T) = \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \quad \tilde{B}^0 V_2 = \begin{bmatrix} B_1 & B_2 \end{bmatrix}, \quad \tilde{q} = \begin{bmatrix} \tilde{q}_1 \\ \tilde{q}_2 \end{bmatrix} = \tilde{U}^T \tilde{q}_3, \quad z_3 = \begin{bmatrix} u \\ v \end{bmatrix}, \] (4.82)
we obtain a transformed DAE of the form
\[ \begin{align*}
\dot{x} &= Ax + B^0 z_3 + \tilde{B}_1 u + \tilde{B}_2 v + p, \\
0 &= (SU^T)x + T_{12} z_1 + \tilde{q}_1, \\
0 &= C_1 x + \tilde{q}_1, \\
0 &= C_2 x + \tilde{q}_2.
\end{align*} \] (4.83a, 4.83b, 4.83c, 4.83d)

Now, it is clear that variable \( u \) is a variable of index two. Variable \( v \) is of index three or higher, i.e. if matrix \( C_2 A B_2 \) is invertible and \( ||C_2 A B_2^{-1}|| \leq M \) for a moderately sized constant \( M \), then \( v \) is of index three.

In the second case the SVD factorization of \( E \) yields
\[ \tilde{U}^T \tilde{E} \tilde{V} = \begin{bmatrix} \Sigma_{11} & \mathbf{0} \\ \mathbf{0} & \Sigma_{22} \end{bmatrix}. \] (4.84)

with \( \Sigma_{11} = \text{diag}(\sigma_i), \; i = 1, \ldots, l, \begin{bmatrix} \Sigma_{22} = \text{diag}(\sigma_i), \; i = l + 1, \ldots, m - r, \end{bmatrix} \begin{bmatrix} \sigma_1 \geq \ldots \geq \sigma_l > 0 \end{bmatrix} \begin{bmatrix} \sigma_{l+1} \geq \ldots \geq \sigma_{m-r} > 0 \end{bmatrix}. \]

Let us now assume that
\[ \sigma_l \gg \varepsilon \gg \sigma_{l+1}. \] (4.85)

With the partitionings (4.82) this results in a system like (4.83). Again, \( u \) is an index two variable. But now, \( v \) is an ill-conditioned index two variable, which means that although \( v \) is formally an index two variable, \( v \) consists of components which will effectively behave like variables of index three or higher. By setting \( (US)^T(B \tilde{V}_2) \tilde{V}_2 \) equal to zero we...
can determine the effective index of $v$ in a similar way as before by studying the matrix $C_2A\bar{B}_2$. For convenience, the system (4.83) is rewritten as
\begin{align}
\dot{x} &= Ax + Bz + Cu + Dv + p, \\ 0 &= Ex + Fz + q, \\ 0 &= Gx + r, \\ 0 &= Hx + s. 
\end{align}

where $z$ is an index one variable since $F$ is invertible, and $u$ is an index two variable, since $GC$ is nonsingular. Differentiation of (4.86d) with respect to time gives
\begin{equation}
0 = H\dot{x} + \dot{s} = H(Ax + Bz + Cu + Dv + p) + \dot{s};
\end{equation}
where $HD = \text{diag} [\varepsilon_i], i = 1, \ldots, m - l - r$ and $\varepsilon \gg \varepsilon_1 \geq \cdots \geq \varepsilon_{m-l-r} \geq 0$, such that $v$ consists only of components which are effectively of index three or higher. If $HD$ were zero and we would differentiate (4.87) once more we would find that $v$ were of index three if $HAD$ were invertible and $\| (HAD)^{-1} \| \leq M$, so $v$ is effectively index three.

Summarizing, we have developed a method by which we can, at least theoretically, separate variables of different index. We can thus determine what the effective index of all component of the algebraic variables is. Naturally, this procedure also works for DAEs of higher index and for more general DAEs such as linear systems with time dependent coefficients and nonlinear systems. Practically, this method may be used to determine in which way the error estimates of the automatic step-size control should be scaled in order to solve DAEs by a code using a variable step-size and order mechanism (cf. Section 6.3). Eventually, it may be used when a code experiences problems due to ill-conditioning of the constraints. After linearizing, the effective index of the variables can be determined and the reduced higher index problem can be solved. In this way the problems due to the ill-conditioning might be circumvented.
5

CONDITIONING OF
DIFFERENTIAL ALGEBRAIC
EQUATIONS OF INDEX TWO

In Chapter 4 we have shown that index one DAEs with an almost singular matrix \( D \) practically behave like DAEs of an index higher than one. For DAEs of index two we derive similar results in this chapter. Furthermore, it appears to be important to study the influence of perturbations of the coefficients. For ODEs and DAEs of index one the sensitivity of the systems with respect to perturbations of the coefficients appeared to be controlled by the stability constants originating from additive perturbations only. However, this does not hold anymore for index two (or higher) DAEs. Therefore, it appears to be important to study the influence of perturbations of the coefficients for DAEs of index two, showing that the stability behaviour of these systems may be (dramatically) altered by perturbations of the coefficients.

5.1 INTRODUCTION

In this chapter we study the conditioning of initial value problems for differential algebraic equations (DAEs) of index two. We will focus on linear systems of Hessenberg form

\[
\begin{align*}
\dot{x}(t) &= A(t)x(t) + B(t)y(t) + p(t), \\
0 &= C(t)x(t) + q(t),
\end{align*}
\]

subject to initial value

\[
x(0) = x_0,
\]

where the solutions \( x : I \to \mathbb{R}^n \) and \( y : I \to \mathbb{R}^m \) with \( I := [0, T] \). The matrix coefficient \( A : I \to \mathbb{R}^{n \times n} \) is continuously bounded on \( I \) and the matrix coefficients \( B : I \to \mathbb{R}^{n \times m} \) and \( C : I \to \mathbb{R}^{m \times n} \) are continuously differentiable on \( I \). In addition, the forcing function \( p : I \to \mathbb{R}^n \) is continuous on \( I \) and \( q : I \to \mathbb{R}^m \) is continuously differentiable on \( I \).
In Section 5.2 we study the conditioning of the index two DAE IVP (5.1) as a result of perturbations of the initial values and the forcing terms. The analysis in Section 5.2 shows that, for stability reasons, one may consider a suitable function space defined by the projector \( P \). Similarly to Section 4.3, the case of an almost singular matrix \( CB \) is studied in Section 5.3. In Section 5.4 the influence of perturbations of the matrix coefficients is studied. We will show that for large \( |P| \), slight perturbations of the coefficients may lead to a dramatic change in the growth behaviour of the solution.

### 5.2 Semi-Explicit Index Two DAEs

In this section we study the conditioning of index two DAE IVPs (5.1). In order to obtain a state equation that describes the behaviour of such a problem, we define the state variable

\[
z := (I - P)x,
\]

where \( P \) is the projector defined by \( P := B(CB)^{-1}C \). This implies that \( z = x + Fq \), with \( F := B(CB)^{-1} \). Clearly \( Cz = 0 \). Therefore, a consistent initial value \( z(0) \) can be defined as follows:

\[
z(0) = x_0 = (I - P(0))u_0, \quad \text{for an arbitrary } u_0 \in \mathcal{R}^n.
\]

This means that

\[
x(0) = x_0 = (I - P(0))u_0 - F(0)q(0),
\]

and therefore,

\[
P(0)x_0 = -F(0)q(0)
\]

and

\[
(I - P(0))x_0 = (I - P(0))u_0.
\]

Hence, only the components \( (I - P(0))x(0) \) of the initial vector \( x(0) \) can be chosen arbitrarily. Using definition (5.2) the DAE (5.1a), (5.1b) can be rewritten as the following equivalent state ordinary differential equation (ODE)

\[
\dot{z} = [(I - P)A - P]z + (I - P)(p - (AF - F)q) = \dot{\hat{A}}z + (I - P)g,
\]

where \( \hat{A} \) and \( g \) are defined as

\[
\hat{A} := (I - P)A - P,
\]

and

\[
g := p - (AF - F)q.
\]
5.2. SEMI-EXPLICIT INDEX TWO DAEs

respectively. We remark that \( z \) does not depend on the derivative of \( q \). Let the fundamental solution matrix \( Z \in \mathbb{R}^{n \times n} \) of ODE (5.7) be defined by

\[
Z = \hat{A}Z, \quad Z(0) = I. \tag{5.10a, 5.10b}
\]

With (5.10b), the solution of ODE (5.7) can be expressed as

\[
z(t) = Z(t)z(0) + \int_0^t Z(t)Z^{-1}(s)(I - P(s))g(s)ds. \tag{5.11}
\]

Since \( z = (I - P)z \), we find from (5.11) that

\[
z(t) = (I - P(t))Z(t)(I - P(0))u_0 + \int_0^t (I - P(t))Z(t)Z^{-1}(s)(I - P(s))g(s)ds. \tag{5.12}
\]

For stability we only have to consider the growth behaviour in the subspace defined by \( \text{range}(I - P) \). Defining the matrix function \( V := (I - P)Z \), the relation (5.12) can be written as

\[
z(t) = V(t)(I - P(0))u_0 + \int_0^t V(t)V^+(s)(I - P(s))g(s)ds. \tag{5.13}
\]

where

\[
V^+ := Z^{-1}(I - P).
\]

Note that \( V \) and \( V^+ \) satisfy

Property 5.1

\[
VV^+V = V. \tag{5.14a}
\]

and

\[
V^+VV^+ = V^+. \tag{5.14b}
\]

We can deduce the following

Lemma 5.2

(i). \( Z \) satisfies

\[
P(t)Z(t)(I - P(0)) = 0. \tag{5.15}
\]

(ii). \( Z^{-1} \) satisfies

\[
(Z^{-1}(I - P))' = -Z^{-1}(I - P)(A(I - P) + P). \tag{5.16}
\]
(iii). The matrix function $Z^{-1}$ satisfies

$$P(0)Z^{-1}(t)(1 - P(t)) = 0. \quad (5.17)$$

Proof.

(i). Equation (5.10a) implies $[PZ] = [\dot{P} - PP]Z$. Since $\dot{P} = PP + PP$, it follows that $[PZ] = PPZ$. Therefore, the matrix function $PZ[1 - P(0)]$ satisfies the differential equation $\dot{U} = PU$, with $U(0) = 0$. Hence, $U(t) = 0$.

(ii). The matrix function $Z^{-1}$ satisfies the differential equation $(Z^{-1})' = -Z^{-1}\dot{A}$. Moreover,

$$Z^{-1}(1 - P') = Z^{-1}\dot{A}(1 - P) - Z^{-1}\dot{P}$$

$$= Z^{-1}(1 - P)A(1 - P) - \dot{P}(1 - P) + \dot{P}$$

$$= Z^{-1}(1 - P)A(1 - P) + PP$$

$$= Z^{-1}(1 - P)(A(1 - P) + \dot{P}).$$

(iii). The matrix function $P(0)Z^{-1}(t)(1 - P(t))$ satisfies the homogeneous differential equation given by (5.16), with $P(0)Z^{-1}(0)(1 - P(0)) = 0$. As a consequence, $P(0)Z^{-1}(t)(1 - P(t)) = 0$.

Hence, (5.15) of Lemma 5.2 implies that

$$(1 - P(t))Z(t)(1 - P(0)) = Z(t)(1 - P(t)). \quad (5.18)$$

From (5.17) of Lemma 5.2 we conclude that

$$(1 - P(0))Z^{-1}(t)(1 - P(t)) = Z^{-1}(t)(1 - P(t)). \quad (5.19)$$

The relations (5.18) and (5.19) combined show

$$(1 - P(t))Z(t)(1 - P(s)) = Z(t)(1 - P(0))Z^{-1}(t)(1 - P(t))$$

$$= Z(t)(1 - P(0))Z^{-1}(t)(1 - P(s))$$

$$= Z(t)Z^{-1}(t)(1 - P(s)). \quad (5.20)$$

The above analysis reveals that the growth behaviour of the state equation is completely governed by $(1 - P(t))Z(t)(1 - P(0))$. Therefore, we define a sort of generalized fundamental solution matrix of (5.7) by

**Definition 5.3** $W(t) = (1 - P(t))Z(t)(1 - P(0))$, with $W(0) = (1 - P(0))$.

We define the generalized inverse $W^+$ of $W$ by
5.3. CONDITIONING FOR ALMOST SINGULAR MATRIX CB

Definition 5.4 \( W^*(t) = (I - P(t))Z^{-1}(t)(I - P(t)) \).

With these definitions both \( W \) and \( W^* \) satisfy property 5.1. This results in the following

Theorem 5.5 The solution \( z \) of the state equation (5.7) can be written as

\[
    z(t) = W(t)(I - P(t))u_0 + \int_0^t W(s)W^*[s](I - P(s))g(s)ds. \tag{5.21}
\]

Proof. The matrix function \( W \) satisfies \( W = \hat{A}W \). Moreover, equation (5.20) yields that \( W(t)W^*[t](I - P(t))g(t) = (I - P(t))g(t) \).

From (5.21), we deduce the following estimates for the growth behaviour of \( z \) and \( x \), i.e.

\[
    \|z\|_\infty \leq \kappa_1 \|(I - P(0))x_0\| + \kappa_2 (\|(I - P)p\|_d + \|(I - P)Afq\|_s + \|(I - P)\hat{F}q\|_s) \tag{5.22}
\]

and

\[
    \|x\|_\infty \leq \kappa_1 (\|(I - P(0))x_0\| + \|F(0)q(0)\|_s) + \kappa_2 (\|(I - P)p\|_d + \|(I - P)Afq\|_s + \|(I - P)\hat{F}q\|_s + \|Pq\|_\infty), \tag{5.23}
\]

respectively, where the conditioning constants are defined as

\[
    \kappa_1 = \max\{\|W(t)\|, t \in I\}, \quad \text{and,} \quad \kappa_2 = \sup\{\int_0^t \|W(s)W^*(s)\|^{1/\tau} ds \mid t \in I\}. \tag{5.24}
\]

where \( \frac{1}{\tau} + 1 = 1 \). The conditioning constants \( \kappa_1 \) and \( \kappa_2 \) the effect of absolute errors due to absolute perturbations of the initial values and the inhomogeneities. The estimates above show that slight perturbations of \( p \) and \( q \) can yield large perturbations in the solution, if \( \kappa_1, \kappa_2 \), or, more importantly, \( \|P\| \) are large. Hence for DAEs not only the conditioning constants, but also the projector \( P \) is very important. Furthermore, the algebraic variable \( y \) satisfies

\[
    y = -(CB)^{-1}((\hat{C} + CA)x + Cp + q) \tag{5.25}
\]

and therefore,

\[
    \|y\|_\infty \leq \|(CB)^{-1}\|((\|C\|_\infty + \|CA\|_\infty)\|x\|_\infty + \|Cp\|_\infty + \|q\|_\infty). \tag{5.26}
\]

5.3 CONDITIONING FOR ALMOST SINGULAR MATRIX CB

In this section we study the conditioning of the IVP (5.1) for an almost singular matrix \( CB \). Subsequently, differentiating the algebraic constraint (5.1b) and substituting the dynamic equation (5.1a) into it results in

\[
    0 = \hat{C}y + CBx + Cp + q. \tag{5.27}
\]
Consequently, the dynamic variable is given by
\[ y = -(CB)^{-1}((C + CA)x + Cp + q). \quad (5.28) \]

Substitution of this equation into the dynamic equation (5.1a) gives the underlying ODE (UODE)
\[ \dot{x} = ((I - P)A - FC)x + (I - P)p - Fq. \quad (5.29) \]

For an almost singular matrix CB the conditioning of the index two IVP (5.1) is given by the following theorem.

**Theorem 5.6** Consider the IVP (5.1). Let CB = εI, (ε ↓ 0). Let CAB be nonsingular with all eigenvalues bounded away from zero. Assume that the DAE is stable. Let the matrix coefficients A, B and C vary slowly. Then the conditioning of this IVP is given by the estimates
\[ \|x\| \leq \kappa_1\|x(0)\| + \kappa_2\|p\| + \|B(CAB)^{-1}\|(\|Cp\| + \kappa_1\|Cp(0)\| + \varepsilon\hat{\kappa}_2\|Cp\|) \]
\[ + \|B(CAB)^{-1}\|(\|q\| + \kappa_1\|q(0)\| + \varepsilon\hat{\kappa}_2\|B(CAB)^{-1}\|\|q\|) \quad (\varepsilon \downarrow 0), \]
\[ \|y\| \leq \hat{\kappa}_2\|CA\||x(0)|| + \hat{\kappa}_2\|Cp\| + \hat{\kappa}_2\|q\| \quad (\varepsilon \downarrow 0). \quad (5.30) \]

The conditioning constants are defined by
\[ \kappa_1 := \max\{\|\exp(-\varepsilon^{-1}BCA)\|, \ 0 \leq t \leq T\}, \]
\[ \hat{\kappa}_1 := \max\{\|\exp(-\varepsilon^{-1}CB)\|, \ 0 \leq t \leq T\}, \]
\[ \kappa_2 := \sup\{\int_0^T \|\exp(-\varepsilon^{-1}BCA(t - \tau))\|d\tau, \ 0 \leq t \leq T\}, \quad \text{and} \]
\[ \hat{\kappa}_2 := \varepsilon^{-1}\sup\{\int_0^T \|\exp(-\varepsilon^{-1}CB(t - \tau))\|d\tau, \ 0 \leq t \leq T\}. \quad (5.32) \]

**Proof.** Let CB = εI, (ε ↓ 0). The equations (5.29) and (5.28) can be rewritten as
\[ x = ((1 - \varepsilon^{-1}BC)A - \varepsilon^{-1}BC)x + (1 - \varepsilon^{-1}BC)p - \varepsilon^{-1}Bq, \quad (5.33) \]
and
\[ y = -\varepsilon^{-1}((C + CA)x + Cp + q), \quad (5.34) \]
respectively. As in the proof of Theorem 4.6 one can see that, in the initial layer, it is sufficiently general to consider the following simpler constant coefficient ODE for x, i.e.
\[ \dot{x} = -\varepsilon^{-1}BCAx + (1 - \varepsilon^{-1}BC)p - \varepsilon^{-1}Bq. \quad (5.35) \]
5.3. CONDITIONING FOR ALMOST SINGULAR MATRIX CB

Using the same matrix T as in (4.46) the matrix BCA can be factorized as

\[
BCA = T \begin{bmatrix} CAB & CAB^4 \\ 0 & 0 \end{bmatrix} T^{-1}. \tag{5.36}
\]

The fundamental solution matrix for equation (5.35) now reads

\[
\Phi(t) = \exp(-\varepsilon^{-1} BCA t). \tag{5.37}
\]

One finds

\[
\exp(-\varepsilon^{-1} BCA t) = T \begin{bmatrix} \exp(-\varepsilon^{-1} CAB t) & (CAB)^{-1} (\exp(-\varepsilon^{-1} CAB t) - 1) CAB^4 \end{bmatrix} T^{-1}, \tag{5.38}
\]

where B denotes a matrix with columns orthonormal to those of B. Similarly to the calculation in (4.51) one finds

\[
\varepsilon^{-1} \int_0^t \exp(-\varepsilon^{-1} BCA(t - \tau)) B f(\tau) \, d\tau = B(CAB)^{-1} (f(t) - \exp(-\varepsilon^{-1} CAB t) f(0)) \\
- B(CAB)^{-1} \int_0^t \exp(-\varepsilon^{-1} CAB(t - \tau)) B q(\tau) \, d\tau, \tag{5.39}
\]

for a function f with suitable dimensions. Using the integral above, x becomes

\[
x(t) = \exp(-\varepsilon^{-1} BCA t)x(0) + \int_0^t \exp(-\varepsilon^{-1} BCA(t - \tau)) p(\tau) \, d\tau \\
- B(CAB)^{-1} \int_0^t \exp(-\varepsilon^{-1} CAB(t - \tau)) C p(\tau) \, d\tau \\
+ B(CAB)^{-1} \int_0^t \exp(-\varepsilon^{-1} CAB(t - \tau)) C p(\tau) \, d\tau \\
- B(CAB)^{-1} \int_0^t \exp(-\varepsilon^{-1} CAB(t - \tau)) q(\tau) \, d\tau. \tag{5.40}
\]

Hence, x can be bounded by (5.30). The dynamic variable y satisfies

\[
y(t) = -\varepsilon^{-1} (C A x(t) + C p(t) + q). \tag{5.41}
\]

Substitution of x from (5.40) results in

\[
y(t) = -\varepsilon^{-1} C A \int_0^t \exp(-\varepsilon^{-1} BCA(t - \tau)) p(\tau) \, d\tau \\
- \varepsilon^{-1} \int_0^t \exp(-\varepsilon^{-1} CAB(t - \tau)) C p(\tau) \, d\tau \\
- \varepsilon^{-1} \int_0^t \exp(-\varepsilon^{-1} CAB(t - \tau)) q(\tau) \, d\tau. \tag{5.42}
\]
Since
\[ CA \exp(-e^{-1}BCA) = \exp(-e^{-1}CAB)CA, \]
the resulting growth behaviour of \( y \) is given by (5.31).

Remark 5.7 The stability assumptions of Theorem 5.6 imply that all eigenvalues \( \lambda_2 \) of \( CAB \) satisfy \( 0 \leq \lambda_2 \leq M \), for a constant \( M \) of moderate size. Therefore, \( \kappa_1 = 1 \) and similarly for \( \kappa_2 \). Further, the conditioning constant \( \kappa_2 \) is of order of magnitude one.

Remark 5.8 Equation (5.30) shows that the matrix \( CAB \) plays an important role in the conditioning of \( x \). Note that this is the same matrix which determines whether the system (5.1) with \( CB = 0 \) has index three.

Remark 5.9 This theorem shows that \( x \) depends on the first derivative of \( q \). Further, \( y \) depends on \( \dot{q} \) and on \( q \) as well.

As in the index one case (see Chapter 4) the \( e^{-1} \) effect is removable in the case that the matrix pencil \( (A, B) \) is regular for \( e \downarrow 0 \), since this implies that the DAE is close to a DAE of higher index. Again for problems with a singular matrix pencil \( (A, B) \) the stability constants might be dominated by \( e^{-1} (e \downarrow 0) \). Such a system is considered in Example 5.12 on page 79.

See example 6.9 for a mechanical example of such a system.

5.4 PERTURBATIONS OF THE COEFFICIENTS

For DAEs of index two or higher it is important to consider not only the influence of perturbations of the initial values and the inhomogeneities, but it is also crucial to study the effect of perturbations of the coefficient matrices. For ODEs and DAEs of index one the perturbations in the homogeneous terms can be regarded as an inhomogeneity of the original IVP. From that analysis one can conclude that a well-conditioned IVP remains reasonably conditioned if the coefficient matrix is perturbed only slightly. For DAEs, however, it is not as straightforward as that, because the state ODEs are derived only after some matrix inversions have been dealt with. This implies that DAEs might be very sensitive to perturbations of the coefficients. In the remainder of this section we shall therefore consider the effect of perturbations of the coefficients \( A, B \) and \( C \) as well. From the original index two DAE (5.1a),(5.1b) we have derived the state ODE (5.7). Its fundamental solution matrix \( Z \) satisfies (5.10), i.e.

\[ \dot{Z} = \dot{A}Z, \quad Z(0) = I. \]

Slight perturbations of the coefficient matrices \( A, B, \) and \( C \) yield the following ODE for the corresponding perturbed fundamental solution matrix \( \hat{Z} \)

\[ \dot{\hat{Z}} = \dot{A}\hat{Z} = (\dot{A} + \delta A)\hat{Z}, \quad \hat{Z}(0) = I. \]
5.4. PERTURBATIONS OF THE COEFFICIENTS

where

$$\delta \hat{A} = (I - P)\delta A - \delta PA - (\delta P^T) + \text{h.o.t.}$$

In the constant coefficient case this gives in first order approximation

$$\delta \hat{A} = (I - P)\delta A - \delta PA.$$ 

Therefore, the perturbation $\delta \hat{A}$ can be large if $A$ or $P$ are large. Hence,

$$\hat{Z}(t) = Z(t) + \int_0^t Z(t)Z^{-1}(s)\delta \hat{A}(s)Z(s)ds$$

$$= Z(t)\left(I + \int_0^t Z^{-1}(s)G(s)Z(s)ds - \int_0^t Z^{-1}(s)H(s)\hat{Z}(s)ds \right). \tag{5.43}$$

Here, the perturbations $G$ and $H$ are defined by

$$G(t) := (I - P(t))\delta A(t)$$

and

$$H(t) := \delta P(t)A(t),$$

respectively. The perturbation $G$ is of concern if the problem is ill-conditioned in the sense that $P$ is skew. The perturbation $H$ may be of concern even when the problem is well-conditioned. Compare relation (5.43) with solution (5.12). Since $(I - P)G = G$, the influence of the second term on the right hand side of (5.43) can be considered to be an additive perturbation, i.e. a perturbation caused by the force function $g$ in equation (5.12). Hence, the contribution of this perturbation is already controlled by the additive perturbations (cf. (5.12)). For the perturbation $H$ this will not hold in general, since $(I - P)H \neq H$. When $H$ is large, $\hat{Z}$ may differ substantially from $Z$. Since $H$ will generally not be in the subspace defined by range$(I - P)$, the difference between $Z$ and $\hat{Z}$ may not be controlled by equation (5.12) implying that the perturbed solution may propagate in the wrong subspace. This implies that the perturbed solution $\hat{Z}$ may differ very much from the unperturbed solution $Z$. From the above, we can draw the following conclusions:

- The perturbation $G$ is of concern if the problem is ill-conditioned in the sense that $P$ is skew. The stability of the solution can change because of $G$. The fundamental solution of the perturbed problem, however, will propagate in the same subspace as the unperturbed solution.

- The perturbation $H$ may be of concern even when the problem is well-conditioned and it may also cause a change in the stability behavior of the solution. However, in this case the perturbed solution might propagate in the wrong directions.

The following theorem shows the possible dramatic influence of perturbations of $B$ and $C$ on the stability behavior of the DAE when $P$ is large in norm.
Theorem 5.10 Consider the constant coefficient index two DAE (5.1a), (5.1b). Assume that $CB = eI$, $e \perp 0$. Let $(I - P)A$ have bounded negative nontrivial eigenvalues. Perturbations of the matrices $B$ and $C$, say $\bar{B} = B + \delta B$ and $\bar{C} = C + \delta C$, may lead to a shift of order $\delta/e$ in the eigenvalues of $(I - P)A$, where $\|\delta B\| = O(\delta)$ and likewise for $\|\delta C\|$.

Proof. The perturbations result in

$$P = P + \delta P,$$

where

$$\delta P = (B + \delta B)(C + \delta C(B + \delta B)^{-1}(C + \delta C) - e^{-1}BC$$
$$= e^{-1}(B + \delta B)[1 + e^{-1}C\delta B + e^{-1}\delta CB + e^{-1}C\delta B]^{-1}(C + \delta C) - e^{-1}BC.$$

Let $\delta B$ and $\delta C$ be small enough so that

$$\max(\|e^{-1}C\delta B\|, \|e^{-1}\delta CB\|, \|e^{-1}C\delta B\|) = \eta, \ \eta \perp 0.$$

Then

$$\delta P = e^{-1}(B + \delta B)[1 - e^{-1}C\delta B - e^{-1}\delta CB + O(\eta^2)][C + \delta C] - e^{-1}BC$$
$$= e^{-1}B\delta C + e^{-1}\delta BC + O(\eta^2/e).$$

The theorem above implies that perturbations of the coefficients of the order $\delta$ may change the stability of the system dramatically when $\delta$ is of the same order of magnitude as $e$.

However, it is not unreasonable to think of much larger perturbations $\delta$ in practice, e.g. if one considers a linearization process for a nonlinear DAE. In the remainder of this section we will consider the influence of perturbations of either one of the coefficient matrices $A$, $B$, or $C$, in order to illustrate the analysis above. First, we study the effect of

perturbations of $A$; so we consider the perturbed DAE

$$\dot{x}(t) = \bar{A}(t)x(t) + B(t)y(t) + p(t),$$
$$0 = C(t)x(t) + q(t),$$

subject to

$$x(0) = x_0,$$

where $\bar{A} = A + \delta A$ and $\bar{x}_0 = x_0 + \delta x_0$. For having consistent initial values $\bar{x}_0$ we need $C(0)x_0 = 0$, i.e. $(I - P(0))\delta x_0 = \delta x_0$. This implies that $\delta x_0$ has to be of the form (5.4).

This leads to the following perturbed state ODE

$$\dot{\bar{x}} = (I - P(\delta A))(I - P)p + (I - P)(\bar{A} - P)q$$
$$= (\bar{A}(t) - P\delta A)x(t) + (I - P)g - \delta AFq + (I - P)\delta AFq.$$
5.4. Perturbations of the Coefficients

Remark 5.11

- The system matrix of (5.45) has a perturbation \((1 - P)\delta A\) as compared to the original state ODE (5.7). This perturbation can be large if \((1 - P)\) is skew. This means that the fundamental solution \(Z\) of (5.45) might exhibit a completely wrong growth behaviour in comparison to \(Z\). For example, it may be possible that a large perturbation \((1 - P)\delta A\) can destroy the stability of the original DAE. In terms of (kinematic) eigenvalues this means that the originally (possibly rather small) negative eigenvalues may result in positive ones.

- Regarding the perturbation \((1 - P)\delta A\) of (5.45) as an inhomogeneity of the original ODE (5.7), it can be seen that this inhomogeneity may be very large if \((1 - P)\) is very large. Hence, a slight perturbation of \(A\) may cause large inhomogeneities which may lead to a large perturbation of \(z\), i.e. \(\|z - \tilde{z}\|\) is large.

- Finally, observe that not only the homogeneous matrix part, but also the inhomogeneous part of the state ODE has changed because of this perturbation of \(A\). So, not only the growth behaviour of \(z\) can be altered dramatically, but also \((1 - P)A\tilde{F}\) can be very different from \((1 - P)\tilde{A}\tilde{F}\).

Example 5.12 Consider the following homogeneous DAE

\[
\begin{align*}
\dot{x} &= \begin{bmatrix} 1/2 & 1/2 \\ 1 & 1/2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} y, \\
0 &= (1 + \varepsilon, -1)x,
\end{align*}
\]

where \(\varepsilon = 10^{-4}\). Here,

\[
P = \varepsilon^{-1} \begin{bmatrix} 1 + \varepsilon & -1 \\ 1 + \varepsilon & -1 \end{bmatrix}
\quad \text{and} \quad (1 - P) = \varepsilon^{-1} \begin{bmatrix} -1 & 1 \\ -(1 + \varepsilon) & 1 + \varepsilon \end{bmatrix}.
\]

Hence \(\|P\| = O(\varepsilon^{-1}) (\varepsilon \to 0)\). The corresponding state ODE is given by

\[
\dot{z} = \frac{1}{2} \varepsilon^{-1} \begin{bmatrix} 1 & -1 \\ 1 + \varepsilon & -(1 + \varepsilon) \end{bmatrix} z.
\]

The eigenvalues of \((1 - P)A\) are 0 and \(-\frac{1}{2}\). Let \(\delta A = \text{diag}[\delta a, 0]\). Then the nonzero eigenvalue of \((1 - P)\tilde{A} = \begin{bmatrix} \frac{1}{2} + \varepsilon^{-1} \delta a \\ \delta a \end{bmatrix}\) which e.g. equals 9.5 when \(\delta a = -10^{-4}\), or, which still equals 1/2 when \(\delta a = -10^{-4}\). In fact, the perturbed DAE is unstable when \(\delta a < -\frac{1}{2}\). In this case, the growth behaviour of the state variables is altered in a dramatic way by this slight perturbation of the matrix \(A\). Hence, a slight perturbation of coefficient matrix \(A\) can alter the solution behaviour dramatically. Further, note that the perturbation of the coefficient matrix of the state ODE equals \(\delta \tilde{A} = (1 - P)\delta A\). This implies that this perturbation is a perturbation of a rather harmless kind, as is already explained in (5.43). Note that the matrix pencil \((\tilde{A}, \tilde{B})\) of this DAE is singular when \(\varepsilon = 0\). The stability constants of this problem are \(O(\varepsilon^{-1}) (\varepsilon \downarrow 0)\), so Theorem 5.6 does not hold for this system.
In appendix C we derive relations between the fundamental solution and the Green function of an unperturbed ODE and its perturbed counterpart. In our DAE case we have a perturbation \((1 - P)\delta A\) of the homogeneous part. When \(\|P\|\) or \(\|A\|\) is large, we see from (4.38) and (C.4) that a slight perturbation \(\delta A\) of \(A\) may cause the fundamental solution of the perturbed state ODE and the unperturbed state ODE to differ dramatically. For the Green function the same holds. As a consequence the growth behaviour of the DAE may be changed in a dramatic way.

Next, we consider the influence of perturbations in the matrix coefficient \(B\) on the solution of the DAE, i.e.,

\[
\dot{x}(t) = A(t)x(t) + \dot{B}(t)y(t) + p(t),
\]

\[
\theta = C(t)x(t) + q(t),
\]

with

\[
\dot{x}(0) = x_0.
\]

where \(\dot{B} = B + \delta B\) and \(x_0 = x_0 + \delta x_0\). We know that the perturbed initial condition has to be a consistent initial condition of the perturbed system. Therefore \(x_0\) has to satisfy \(C(0)\delta x_0 = 0\), which implies \((I - P(0))\delta x_0 = \delta x_0\); hence \(x_0\) should be of the form (5.4).

The corresponding perturbed state ODE can be written as

\[
\dot{z} = (I - \dot{P})A - \dot{P}z + (I - \dot{P})g
\]

\[
= [A - (I - P)\delta B(CB)^{-1}CA - ((I - P)\delta B(CB)^{-1}C)]z
\]

\[
+ (I - P)(1 - \delta B(CB)^{-1}C)g
\]

\[
- (I - P)(A(I - P)\delta B(CB)^{-1} - ((I - P)\delta B(CB)^{-1})q),
\]

because \(\dot{P} = P + (I - P)\delta B(CB)^{-1}C\).

**Remark 5.13**

- As before, the resulting perturbation of the homogeneous part of the state ODE (5.45) might be large if \((I - P)\) is large. This implies that the growth behaviour of the perturbed problem might be completely different from the growth behaviour of the original problem.

- The fundamental solutions \(Z\) and \(\dot{Z}\) and the Green functions \(G\) and \(\dot{G}\) might differ dramatically if \(\|P\|\) is large (cf. appendix C).

- The perturbations of the inhomogeneous part of the state ODE can be very large if \(\|P\|\) is large, which means that \(\|z - \dot{z}\|\) can be very large.
As a consequence, the perturbed solution $\tilde{x}$ satisfies
\[
\tilde{x} = \tilde{z} + P\tilde{x} = \tilde{z} - Fq - (I - P)\tilde{A}(CB)^{-1}q. \tag{5.48}
\]
Hence, due to the arguments above x and $\tilde{x}$ might differ dramatically.

**Example 5.14** Again, consider Example 5.12. Now, the coefficient matrix $B$ is perturbed slightly, say
\[ B := B + \delta_{1,1}J. \]

Then
\[
(I - \tilde{P}) = I - B(CB)^{-1}C = \frac{1}{\delta - \delta} \begin{bmatrix} -1 + \delta & 1 \\ -1 + \delta & 1 + \delta \end{bmatrix}
\]

\[
\tilde{A} = (I - \tilde{P})A = \frac{1}{2(\delta - \delta)} \begin{bmatrix} \delta - 1 & 1 + 2\delta \\ 1 + \delta - 1 & (1 + \delta)(1 + 2\delta) \end{bmatrix},
\]

with nonzero eigenvalue $\delta = 3.55556$. For $\delta = 5.10$, this nonzero eigenvalue of the perturbed system is equal to 1.667776, which implies that the originally stable system is perturbed into an unstable one by a small perturbation of the matrix $B$. Note further, that $(I - P)\tilde{A} = \tilde{A}$. This means that we deal with a perturbation of the state ODE which is of the same kind as the second term of (5.43) and from (5.43) we know that this perturbation may be controlled already.

Next we study the effect of perturbations of the matrix coefficient $C$ on the solution of the DAE system, i.e. we consider the perturbed problem
\[
\begin{align*}
\dot{\tilde{x}}(t) &= A(t)\tilde{x}(t) + B(t)\tilde{y}(t) + p(t), \tag{5.49a} \\
\theta &= C(t)\tilde{x}(t) + q(t). \tag{5.49b}
\end{align*}
\]
subject to
\[
\tilde{x}(0) = \tilde{x}_0, \tag{5.49c}
\]

where $\tilde{C} = C + \delta C$ and $\tilde{x}_0 = x_0 + \delta x_0$. Now, for a consistent initial condition of the perturbed system $\tilde{x}_0$ should satisfy $\tilde{C}(0)\delta x_0 = 0$. For the first order terms this means that $C(0)x_0 + \delta C(0)x_0 = 0$. Again, this implies that $\delta x_0$ should be of the form (5.4). In this case one finds for the perturbed projector $\tilde{P}$
\[
P = P + F\tilde{C}(I - P).
\]

Then, the corresponding perturbed state ODE can be written as
\[
\begin{align*}
\dot{\tilde{z}} &= (I - \tilde{P})A - \tilde{P}\tilde{z} + (I - \tilde{P})\tilde{g} \\
&= (\tilde{A} - F\tilde{C}(I - P)A - (F\tilde{C}(I - P)) \tilde{g} \\
&\quad + (I - F\tilde{C})(I - P)g + (I - P)(AF\tilde{C} - (F\tilde{C})')q. \tag{5.50}
\end{align*}
\]
In this case, the same remarks hold as before. Now, we find the following expression for \( \tilde{x} \)

\[
\tilde{x} = \dot{z} + \tilde{\Phi} \tilde{x} = \dot{z} - Fq - \Phi \delta C_F q
\]

and again, the error \( \|x - \tilde{x}\| \) might become very large. Next

**Example 5.15** Consider the DAE

\[
\begin{align*}
\dot{x} &= \begin{bmatrix} \lambda & 0 \\ 0 & \mu \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} y + p, \\
0 &= \begin{bmatrix} 1 & 1 \end{bmatrix} x + q.
\end{align*}
\]

The corresponding state ODE is

\[
\dot{z} = \frac{1}{2} \begin{bmatrix} \lambda & -\mu \\ -\lambda & \mu \end{bmatrix} z + \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} p + \frac{1}{4} (\mu - \lambda) \begin{bmatrix} 1 \\ 1 \end{bmatrix} q.
\]

The eigenvalues of \((I - \Phi)A\) are 0 and \( \frac{1}{2}(\lambda + \mu) \), respectively. Let \( \delta C = [a \ s] \), then

\[
(I - \Phi - \delta \Phi)A = \frac{1}{2 + \varepsilon} \begin{bmatrix} \lambda & -\mu \\ -\lambda(1 + \varepsilon) & \mu(1 + \varepsilon) \end{bmatrix},
\]

with eigenvalues 0 and \( \frac{1}{2(1 + \varepsilon)}(\lambda + \mu(1 + \varepsilon)) \), respectively. A typical stable situation (for \( \varepsilon = 0 \)) is \( \lambda \approx -\mu \). The perturbed problem has nontrivial eigenvalue \( \frac{\mu}{2(1 + \varepsilon)} \). So, we may expect difficulties if \( |\lambda| > 1 \). In particular if \( \lambda > 1 \) the DAE, a slight perturbation of the stable problem, has resulted in a very unstable one.

In the following important example, we consider a nearly index three DAE.

**Example 5.16** Consider the homogeneous DAE

\[
\begin{align*}
\dot{x} &= \begin{bmatrix} 1/2 & 1 \\ 1 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} y, \\
0 &= \begin{bmatrix} 1 + \varepsilon & -1 \\ -1 & 1 + \varepsilon \end{bmatrix} x,
\end{align*}
\]

where \( \varepsilon = 10^{-4} \). The corresponding state ODE is given by

\[
\dot{z} = (1 + \frac{1}{2} \varepsilon^{-1}) z^{-1} \begin{bmatrix} 1 & -1 \\ 1 + \varepsilon & -(1 + \varepsilon) \end{bmatrix} z.
\]

The eigenvalues of \((I - \Phi)A\) are 0 and \(-1(1 + \frac{1}{2} \varepsilon^{-1}) \approx -5199\), respectively. Hence, the system is extremely stable for \( \varepsilon \downarrow 0 \). Let \( \delta C = [a \ s] \). Then

\[
\tilde{A} = (1 - \tilde{\Phi})A = \frac{1}{2(\varepsilon + \delta)} \begin{bmatrix} 1 - \delta & 2(\delta - 1) \\ (1 + \varepsilon) & -2(1 + \varepsilon) \end{bmatrix}.
\]
with eigenvalues 0 and $\frac{2+\pi \pm \sqrt{1+4\pi^2}}{2\pi}$. For $\delta = -2\pi + 1$, the nonzero eigenvalue of the perturbed coefficient matrix $\hat{A}$ is equal to $5_{10}$. This means that a DAE which was extremely stable is perturbed into an extremely unstable DAE by a slight perturbation of the coefficient matrix $C$. Here, the direction of the mode corresponding to the nonzero eigenvalue is perturbed into $[1 - \delta, 1 + \epsilon]^T$. Hence, the nontrivial solution modes are altered in a dramatic way, viz. the growth of this mode has been changed severely, while simultaneously the direction of this mode is altered. Hence, perturbations of type $H$ may lead to dramatic results. Note that the matrix pencil $(\hat{A}, \hat{B})$ of this DAE is regular for $\epsilon = 0$, implying (cf. Theorem 5.6) that the stability constants of this system are bounded. □
The study of numerical methods for solving DAEs of higher index is the subject of this chapter. In Chapter 3 we have shown that numerical methods for solving stiff ODEs are in general only convergent for index one systems and a loss of the approximation order may occur in the algebraic variables. Fortunately, BDF methods converge for some important classes of DAEs, e.g. systems in Hessenberg forms. However, BDF codes, using variable step-sizes and variable order, still do not work properly. The equations of motion for multibody systems are generally index three. In order to solve these equations by BDF codes one has to reduce the index to one by successive differentiations of the constraints. This, however, results in numerical solutions drifting away from the constraints, as will be shown in Section 6.1. To alleviate this effect of drifting away from the constraints, special techniques for solving higher index DAEs, and equations of motion for multibody systems in particular, have to be developed. In Section 6.1 such methods are studied. Discretization of higher index DAEs by BDF methods yields ill-conditioned iteration matrices. The effects of this ill-conditioning on the numerical solution of DAEs is analysed in Section 6.2; we show that it can affect the numerical solution badly. For DAEs that are nearly higher index, as considered in the Chapters 4 and 5 this effect turns out to be even worse. A method is given to reduce the effects of ill-conditioned iteration matrices on the approximation of the solution of DAEs of (nearly) higher index. The consequences for variable step-size and variable order BDF codes are elaborated in Section 6.3. We will show that these ill-conditioning effects can be remedied in a way allowing for the numerical solution of DAEs in higher index form directly. In Section 6.4 applications to multibody systems are shown. In particular, the numerical solution of these mechanical systems using a projector technique appears to be preferable.
6.1 SOLUTION TECHNIQUES FOR HIGHER INDEX DAEs

In this section integration schemes for the equations of motion generated by multibody systems are discussed. In Sections 3.2 and 3.3 the numerical solution of DAEs by multistep and Runge-Kutta methods is studied. These methods can generally not be applied to higher index \( \nu \geq 2 \) DAEs, since most of them are convergent for index one problems only. Another problem is the loss of accuracy for the algebraic variables. This causes standard techniques of error estimation and step-size selection to fail. These problems can be avoided by differentiating the DAEs \( \nu - 1 \) times with respect to time. Such index-reduced system can then be solved numerically by multistep or Runge-Kutta methods, for example. This approach gives rise to two problems. First, the numerical solution of the index-reduced system does not fulfill the original constraints every step: hence due to error propagation, the numerical solution tends to drift away from the algebraic constraints (cf. [27]). Secondly, the stability of the DAE with respect to perturbations in the solution may change due to the index transformation (cf. [28]). To show the effect of this index reduction we again consider the pendulum studied in the Examples 2.1 through 2.3.

Example 6.1 Let us choose \( l = 1, m = 1 \) and \( g \) such that the pendulum swings with a period \( T = 2 \), i.e. \( g = 4/3 \cdot 13.75 \)\(^1\). The initial positions \( x(0) = 0, y(0) = 1 \) and \( \varphi(0) = \pi/2 \), together with the initial velocities \( \dot{x}(0) = y(0) = \dot{\varphi}(0) = 0 \) are consistent as can easily be seen. We compute the numerical solution over 50 periods with a toler-

![Position Constraints](image1.png)

![Velocity Constraints](image2.png)

Figure 6.1 Resulting drift for \( \text{TOL} = 10^{-4} \). The dotted lines represent the absolute deviations from the position constraint \( x - l \cos \varphi = 0 \) (denoted by \( \delta_p \)) and its derivative (denoted by \( \delta_{\dot{p}} \)), whereas the deviations from the position constraint \( y - l \sin \varphi = 0 \) and its derivative are represented by the solid lines.

\[ \text{TOL} = 10^{-4} \] Figure 6.1 shows that the numerical solution of the reduced DAE drifts away from the constraints. The resulting global errors at \( t = 100 \) are given by \( e_x := |x(t) - x_a| = 0.16 \cdot 10^{-1} \), \( e_y = 0.83 \), \( e_{\varphi} = 0.99 \), \( e_{\dot{x}} = 0.20 \cdot 10^2 \) and \( e_{\dot{\varphi}} = 0.30 \cdot 10^2 \).

\(^1\) We used \( g = 4/3 \cdot 13.7505371386407457 \) which has been computed using elliptic integrals.
where $\lambda_i$ ($i = 1, 2$) denote the Lagrange multipliers (cf. Example 2.2). Hence, the numerical solution is useless.

In the following, several methods especially designed for solving the equations of motion for multibody systems are discussed.

### 6.1.1 Regularization Methods

The regularization of a DAE can be interpreted as the introduction of a small parameter in the DAE such that the solution of the perturbed system approaches the solution of the original DAE as the parameter tends to zero. In this subsection several approaches which can be interpreted as regularization methods are described.

The oldest regularization method for alleviating the problem of drifting away from the constraint $\phi = 0$ ((3.19c)) was introduced by Baumgarte (cf. [7]). He introduced stabilizing control terms into the index one DAE (3.19a), (3.19b), (3.21). Instead of the acceleration constraints, a linear combination of the acceleration, the velocity and the position constraints

$$\ddot{\phi} + 2\alpha \dot{\phi} + \beta^2 \phi = 0, \quad \text{(6.1)}$$

with parameters $\alpha$ and $\beta$ is used. The resulting DAE

$$\begin{align*}
\dot{q} &= v, \\
M(q, t)v &= g(q, v, t) - \phi^2 \lambda, \\
0 &= \ddot{\phi} + 2\alpha \dot{\phi} + \beta^2 \phi.
\end{align*} \quad \text{(6.2)}$$

has index one. The Baumgarte parameters $\alpha$ and $\beta$ in (6.1) are chosen such that $\phi = 0$ is a stable solution of (6.1). This yields $\alpha > 0$. Often one chooses $\alpha = \beta$, which corresponds to the aperiodic limit case (or the critical damping condition). The index one DAE (6.2) can then be solved by integration techniques that are convergent for index one DAEs. Numerical evidence shows that the drift-off from the algebraic constraint becomes essentially weaker than in the original case. However, a problem is the choice of the Baumgarte parameters $\alpha$ and $\beta$. Choosing them too large results in a stiff system, because extraneous eigenvalues are introduced into the system. Choosing the parameters too small minimizes the stabilization effect. Ascher and Petzold (cf. [4]) even report a problem where the Baumgarte parameter should approach infinity to effectively reduce the drift.

**Example 6.2** Application of Baumgarte’s approach to the pendulum of Example 6.1 results in the drift as shown in Figure 6.2. Hence, using this method reduces the drift considerably (cf. Figure 6.1). The resulting global errors at $t = 100$ are $e_x = 0.31 \cdot 10^{-4}$, $e_{\dot{x}} = 0.20 \cdot 10^{-3}$, $e_{\theta} = 0.29 \cdot 10^{-3}$, $e_{\dot{\theta}} = 0.48 \cdot 10^{-2}$ and $e_{\lambda} = 0.78 \cdot 10^{-5}$, where the errors are defined by $e_i := |x(t) - x_i|$ and likewise for the other errors. Figure 6.3 shows that this method becomes more and more expensive as $\alpha = \beta$ becomes larger.
Figure 6.2  Resulting drift using Baumgarte’s approach with \( a = \beta = 6 \), which seems to be a more or less optimal choice for \( a \), for \( \text{TOL} = 10^{-4} \). The dotted lines represent the absolute deviations from the position constraint \( x - L \cos \varphi = 0 \) (denoted by \( \delta_p \)) and its derivative (denoted by \( \delta_t \)), whereas the deviations from the position constraint \( y - f \sin \varphi = 0 \) and its derivative are represented by the solid lines.

Figure 6.3  Number of steps as function of the Baumgarte parameter \( a = \beta \), for \( \text{TOL} = 10^{-4} \) and \( t = 100 \). The straight line denotes the number of steps needed for solving the projection method (6.12) (see Subsection 6.1.3).

Other regularization techniques were proposed by Lütstedt, Knorrechild and Hanke and have been compared by Eich and Hanke (cf. [22]), who showed that these methods are actually very similar. Löstedt (cf. [22]) introduced penalty functions which lead to the equation

\[
M \dot{q} = g - \varepsilon^{-1} \kappa. \tag{6.3}
\]

Applying Knorrechild’s approach yields

\[
M \dot{q} = g - \phi \dot{\lambda}, \quad 0 = \phi (q + (\varepsilon + \mu) \dot{q} + \mu \varepsilon \lambda^{-1} (g - \phi \dot{\lambda})), \tag{6.4}
\]

while Hanke’s regularization technique results into

\[
q = v + \mu \dot{v}, \\
M \dot{v} = g(q, v, r) - \phi \dot{\lambda}, \\
0 = \phi (q + \varepsilon \dot{q} + \mu \varepsilon \lambda^{-1} (g - \phi \dot{\lambda})). \tag{6.5}
\]
6.1. Solution Techniques for Higher Index DAEs

Eich and Hanke showed that the methods of Baumgarte, Knorr, and Hanke differ in higher order terms in \( \varepsilon \) and \( \mu \) only.

6.1.2 Generalized Coordinate Partitioning

Using a differential geometric approach (cf. [71]) DAEs can be interpreted as differential equations on manifolds. Therefore, DAEs can be parameterized, at least locally, as differential equations on manifolds. The constraints can be used to define this local parameterization, which defines a local bijective correspondence between the state variable and the variable on the parameter space. Węhage and Haug (cf. [80]) and Rheinboldt (cf. [71]) developed differential geometric techniques to determine this local coordinate system, where the ODE is integrated by standard methods. It is illustrative to describe the generalized coordinate partitioning method developed by Węhage and Haug (cf. [80]).

Consider the equations of motion (cf. also Subsection 3.1.5)

\[ \dot{q} = v, \quad (6.6a) \]
\[ M(q, r)v = g(q, v, t) - \phi_q \lambda, \quad (6.6b) \]
\[ \theta = \phi(q, t), \quad (6.6c) \]

together with the velocity and acceleration constraints

\[ \phi_q q = v \quad (6.7) \]

and

\[ \phi_q \ddot{q} = \gamma. \quad (6.8) \]

respectively. The Jacobian matrix \( \phi_q = \frac{\partial \phi}{\partial q} \) has full row rank \( m \), say. So, there is at least one nonsingular submatrix of \( \phi_q \) of rank \( m \). The vector \( q \) can be partitioned as

\[ q = \begin{bmatrix} u \\ w \end{bmatrix}. \quad (6.9) \]

such that \( \phi_u \) is the aforementioned nonsingular submatrix of \( \phi_q \) corresponding to \( u \). Therefore, a partitioning of \( \phi_q \) has to be carried out by e.g. Gauss-Jordan reduction with complete pivoting or by SVD or QU factorization. Then, \( u \) denotes the dependent generalized coordinates and \( w \) denotes the independent generalized coordinates. Using LU factorization results in independent coordinates \( w \) that are part of the generalized coordinates \( q \), whereas the use of SVD factorization implies that the independent coordinates are a linear combination of the generalized coordinates. Then the implicit function Theorem assures that there exists a twice differentiable function \( h = h(w, r) \) such that

\[ u = h(w, r), \quad (6.10) \]

is the solution of (6.6c) for \( u \) as function of \( w \) and \( t \). By relation (6.10) the equations of motion can be rewritten in terms of the independent generalized coordinates \( w \). After
elimination of the Lagrange multipliers, the equations of motion can be expressed as an ODE for the coordinates $w$, i.e.

$$
\ddot{w} = s,
\tilde{M}(w, t)\dot{s} = \tilde{g}(w, s, t).
$$

This ODE can be solved numerically by standard methods. Given the complexity of the matrix $\tilde{M}$ and the vector $\tilde{g}$ of (6.11) as functions of $w$ and $s$, direct discretization of (6.11) would be very complicated and impracticable. Therefore, one solves $\ddot{q} = [\ddot{q}^T, \dot{\tilde{q}}^T]^T$ from the DAE (6.6a), (6.6b), (6.7). Now $w$ can be integrated by any explicit ODE solver to find $w$ and $\dot{w}$. Afterwards $u$ and $\dot{u}$ can be obtained by solving (6.6c) and (6.7), respectively. This can be continued as long as the partitioning (6.9) does not need to be changed. However, if $\phi_\alpha$ becomes ill-conditioned, then the generalized coordinates $q$ should be repartitioned, which is a major disadvantage of this method. Huang and Yen (cf. [47]) also developed an implicit DAE solver based on generalized coordinate partitioning.

### 6.1.3 Projection Methods

Drift-off from the constraints can be avoided by numerically solving a DAE with reduced index, combined with a projection such that the original constraints are satisfied. Eich et al. (cf. [21]) have shown that these projections can be divided into two classes, depending on whether they rely on stabilizing projections of position and velocity variables (coordinate projection methods) or on projections of residuals of the differential equations (derivative projection methods).

Gear et al. (cf. [34]) have introduced stabilizing Lagrange multipliers $\mu$, say, to simultaneously reduce the index and satisfy the position constraint and the velocity constraint

$$
\dot{q} = v - \phi_\alpha^R\mu,
M(q, t)\dot{v} = g(q, v, t) - \phi_\alpha^R\lambda,
0 = \phi(q, t),
0 = \phi(q, t)v - v.
$$

A solution of (6.12) exists only if the additional stabilizing multipliers $\mu$ satisfy $\mu = 0$. Hence, (6.12) and the original system have the same solution. DAE (6.12) has index two and can be integrated numerically. We have applied this projection method to the pendulum of Example 6.1.

**Example 6.3** For a tolerance $\text{TOL} = 10^{-4}$ the deviation of the constraints is smaller than $0.44 \cdot 10^{-5}$. The resulting errors at $t = 100$ are $e_x = 0.13 \cdot 10^{-5}$, $e_y = 0.16 \cdot 10^{-5}$, $e_\phi = 0.16 \cdot 10^{-2}$, $e_\alpha = 0.29 \cdot 10^{-1}$ and $e_\mu = 0.71 \cdot 10^{-4}$. Figure 6.3 shows that the number of steps\(^4\) taken in this method is considerably smaller than in Baumgarte’s approach.

\(^4\)In the computation we used the scaling of the local error estimates and the scaling of the error estimates in the Newton iterations, as developed in Section 6.3
6.1. SOLUTION TECHNIQUES FOR HIGHER INDEX DAEs

Führer and Leimkuhler (cf. [29]) have extended this idea to the index one system. Therefore, two additional multipliers \( \mu \) and \( \tau \), say, have to be introduced to satisfy the position constraint, the velocity constraint and the acceleration constraint

\[
\begin{align*}
\dot{q} &= v - \phi^T_2 \mu - [v^T \phi_1] \tau, \\
M(q, t) \ddot{v} &= g(q, v, t) - \phi^T_2 \lambda - \phi^T_1 \tau, \\
\theta &= \phi(q, t), \\
\theta &= \phi_q(q, t) q - v, \\
\theta &= \phi_q(q, t) q - y.
\end{align*}
\]  \tag{6.13}

This DAE has index two and the index two variables \( \mu \) and \( \tau \) have to be equal to zero to assure existence of a solution. The last two approaches can be interpreted as derivative projections onto state space forms. One can show that the coordinate partitioning methods (cf. [80]), the differential geometric approach (cf. [69]) and methods using overdetermined differential algebraic equations (which will be discussed in the next subsection) can be seen as derivative projection methods.

Coordinate projection methods project the computed solution of the UODE (cf. Definition 3.7) or the index one DAE onto the constraint manifold. Shampine (cf. [75]) was the first person to describe this technique for one step methods and Eich (cf. [20]) has given a convergence proof of this method in the context of multistep methods.

6.1.4 OVERDETERMINED DIFFERENTIAL ALGEBRAIC EQUATIONS

Another approach has been followed by Führer (cf. [27]). To circumvent the problem of drifting away from the constraints, not only the position constraints (3.19c), but also the velocity (3.20) and the acceleration constraints (3.21) are used. Together with the dynamic equations (3.19a) and (3.19b), this results in an overdetermined differential algebraic equation (ODAE)

\[
\begin{align*}
q &= v, \\
M(q, t) \dot{v} &= g(q, v, t) - \phi^T_2 \lambda, \\
\theta &= \phi(q, t), \\
\theta &= \phi_q(q, t) q - v, \\
\theta &= \phi_q(q, t) q - y.
\end{align*}
\]  \tag{6.14a-6.14e}

This ODAE has index one. It has a unique solution for consistent initial values. This solution is identical to the solution of the original index three DAE (6.14a),(6.14b),(6.14c).

Discretizing the ODAE by e.g. a BDF method results in

\[
\begin{bmatrix}
\phi(E) q_n - v_n \\
M(q_n, t_n) \rho(E) v_{n+1} - g(q_n, v_n, t_n) + \phi_q(q_n, t_n)^T \lambda_n \\
\phi_q(q_n, t_n) \rho(E) q_{n+1} - \gamma_n \\
\phi_q(q_n, t_n) \rho(E) v_{n+1} - \gamma_n
\end{bmatrix} = 0. \tag{6.15}
\]
where $\rho$ (cf. (3.31)) is the generating polynomial of the $k$-step BDF method and $\ell$ is the shift operator (see Section 3.2). However, the discretized version (6.15) does not have a unique numerical solution. Therefore, this system has to be solved in a least squares sense. In other words, the numerical solution must satisfy the equations (6.15) in a generalized inverse sense. Führer and Leimkuhler (cf. [29]) have shown that there exists a generalized solution, which is (in the linearized case) numerically equivalent to the reduction to state space form, cf. (3.27), of the linearized equations of motion. Therefore it is called the SSF-solution. Moreover, using the SSF-solution of (6.15) within a BDF method is equivalent to solving the stabilized problem (6.13) with the same BDF method.

### 6.2 THE ITERATION MATRIX

In Chapter 4 we have shown that semi-explicit DAEs of index one may be arbitrarily close to DAEs of higher index $\nu$, i.e. $\nu \geq 2$. In fact, such DAEs effectively behave like DAEs of higher index. It is well known that (semi-explicit) DAEs of higher index are ill-posed in a sense, since they depend on the $(\nu - 1)^{th}$ derivative of the forcing function. Hence, when solving index one DAEs which are nearly higher index smoothness problems may arise, because the solution may depend on derivatives of the forcing function which are of higher order than expected. Since such a DAE is arbitrarily close to a DAE of higher index the stability of the problem will be governed by the associated higher index system. So, the steady state solution of the ill-conditioned problem is governed by the reduced system of higher index. As a consequence of the ill-posedness due to differentiating, this leads to another problem, viz. the conditioning of the iteration matrix originating from the numerical method. It is well known that applying numerical ODE methods, such as BDF and implicit Runge-Kutta (IRK), to DAEs of higher index results in ill-conditioned iteration matrices. In fact, the conditioning constant of the iteration matrix resulting from discretization of an index $\nu$ DAE using backward differences is $O(h^{-\nu})$ (cf. [68]) in the so-called direct approach (cf. (3.36)), where $h$ is the step-size. In that case, the conditioning constant can be reduced to $O(h^{-\nu+1})$ by scaling. Below we shall show that the conditioning constant of the resulting iteration matrix is $O(h^{-\nu+1})$ (cf. [43]) in the indirect approach (cf. (3.37)) and therefore, errors in the initial values or numerical rounding errors are amplified by factors $h^{-\nu+1}$. The previous remark holds for DAE systems which themselves are well-conditioned.

#### 6.2.1 Conditioning of the Iteration Matrix

In this subsection we elaborate on the conditioning aspects of the iteration matrix for semi-explicit DAEs. Consider the index one DAE first

\[ x' = Ax + By + p \]  
\[ 0 = Cx + Dy + q \]
6.2. The Iteration Matrix

In the indirect approach, numerical ODE methods for the approximation of the solution of the DAE above will essentially involve an iteration matrix of the following form

$$\hat{h}J_I := \begin{bmatrix} I - \hat{h}A & -\hat{h}B \\ C & D \end{bmatrix},$$

(6.17)

where e.g. in the case of multistep methods $\hat{h} := h\bar{h}_0/a_0$. The inverse of $\hat{h}J_I$ can be computed through LU-decomposition, i.e.

$$\hat{h}J_I = LU := \begin{bmatrix} I & 0 \\ C(I - \hat{h}A)^{-1} & I \end{bmatrix} \begin{bmatrix} I - \hat{h}A & -\hat{h}B \\ C(I - \hat{h}A)^{-1}B + D \end{bmatrix}.$$  

(6.18)

For $U^{-1}$ we need the inverse of the matrix $\hat{h}C(I - \hat{h}A)^{-1}B + D$.

Let us first assume that the matrix $D$ is such that

$$(\hat{h}C(I - \hat{h}A)^{-1}B + D)^{-1} = D^{-1} - \hat{h}D^{-1}CBD^{-1} + O(\hat{h}^2).$$  

(6.19)

This implies that

$$(\hat{h}J_I)^{-1} = \begin{bmatrix} I + \hat{h}(A - BD^{-1}C) & -\hat{h}BD^{-1} \\ -D^{-1}C(I + \hat{h}(A - BD^{-1}C)) & (I - \hat{h}D^{-1}CBD)D^{-1} \end{bmatrix} + O(\hat{h}^2).$$  

(6.20)

Notice the important role of the matrices $D^{-1}C$ and $A - BD^{-1}C$ in the equation above. Hence, rounding errors proportional to the machine constant $\eta$ be introduced both in $x$ and in $y$, while solving this linear system. This means that such a DAE behaves as well conditioned as an ODE. Here and in the sequel the latter has to be understood in an absolute (i.e. not relative) sense. If, on the other hand, $|D^{-1}|$ is not small, problems arise.

For simplicity take $D = A$, $\{e \to 0\}$ (see also Section 4.3), then we find for small fixed $\hat{h}$ that

$$(\hat{h}C(I - \hat{h}A)^{-1}B + D)^{-1} = \hat{h}^{-1}(CB)^{-1}(I - \hat{h}CAB(CB)^{-1}) + O(\hat{h}) + O(e/\hat{h}^2), \{e \to 0\},$$

(6.21)

when the matrix $CB$ is well-conditioned and bounded away from zero. We thus find

$$(\hat{h}J_I)^{-1} = \begin{bmatrix} (I - P)[(I + \hat{h}A)(I - P)] & (I + \hat{h}(I - P)A)B(CB)^{-1} \\ -(\hat{h}^{-1}(CB)^{-1}C(I + \hat{h}A)(I - P)) & \hat{h}^{-1}(CB)^{-1}(I - \hat{h}CAB(CB)^{-1}) \end{bmatrix} + h.o.t., \{e \to 0\},$$

(6.22)

where the projector $P$ is defined by $P := B(CB)^{-1}C$, cf. also Section 5.2. This implies that rounding errors proportional to $\eta$ and $\eta^2\hat{h}^{-1}$ may be introduced in the variables $x$ and $y$, respectively. Here and in the sequel we assume that the (approximate) variables have a moderate norm, so that absolute and relative errors are qualitatively the same. Hence, we shall consider absolute "machine" rounding errors $\delta$ rather than relative rounding errors $O(\eta)$ in our examples. The matrix $(\hat{h}J_I)^{-1}$ (cf. (6.22)) shows the ill-posedness, due to differentiation of the forcing function, of an index one DAE which is close to a DAE of higher index (cf. Subsection 4.3).
Example 6.4 Consider the index one DAE
\[ \dot{x} = -y + \sin t, \]
\[ 0 = x - \cos t, \]
subject to the initial condition \( x(0) = 2. \) On the interval \([0, 10^{-3}]\) we obtain from discretization with Euler backward Table 6.1, where \( e_x := |x(t) - x_h|, \) \( e_y := |y(t) - y_h| \) and drift := \( |x - \dot{x}_h - \cos t_h|, \) i.e. the deviation from the constraint, for \( nh = T = 10^{-3}. \) In order to show the influence of the rounding errors we introduce artificial absolute rounding errors \( \delta, \) say, with \( \delta \in [4 \cdot 10^{-5}, 6 \cdot 10^{-5}], \) into the system, where \( \varepsilon \) was taken equal to \( 10^{-6}. \) It is obvious that errors proportional to \( \delta \) are introduced into the state vari-

<table>
<thead>
<tr>
<th>( h )</th>
<th>( e_x )</th>
<th>( e_y )</th>
<th>drift</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-4} )</td>
<td>( 0.49 \cdot 10^{-4} )</td>
<td>( 0.75 \cdot 10^{-1} )</td>
<td>( 0.49 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( 10^{-3} )</td>
<td>( 0.55 \cdot 10^{-4} )</td>
<td>( 0.13 \cdot 10^{-1} )</td>
<td>( 0.55 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>( 0.54 \cdot 10^{-4} )</td>
<td>( 0.14 \cdot 10^{-1} )</td>
<td>( 0.54 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( 10^{-3} )</td>
<td>( 0.55 \cdot 10^{-4} )</td>
<td>( 0.30 \cdot 10^{-1} )</td>
<td>( 0.58 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( 10^{-5} )</td>
<td>( 0.55 \cdot 10^{-4} )</td>
<td>( 0.10 \cdot 10^{-1} )</td>
<td>( 0.56 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( 10^{-6} )</td>
<td>( 0.55 \cdot 10^{-4} )</td>
<td>( 0.35 \cdot 10^{-1} )</td>
<td>( 0.58 \cdot 10^{-4} )</td>
</tr>
</tbody>
</table>

Table 6.1 Results for the index two DAE of Example 6.5 showing the \( h^{-1} \) effect in the algebraic variable \( y. \)

able \( x. \) Note that from the first three rows of Table 6.1 it appears that the numerical approximation of the algebraic variable \( y \) has errors proportional to \( \delta h^{-1}. \) In the last three rows, however, the resulting errors are proportional to \( \delta e^{-1}; \) this is explained in Subsection 6.2.2.

For a Hessenberg form DAE of index two
\[ \dot{x} = Ax + By + p, \]
\[ \theta = Cx + q, \] (6.24)

where \( CB \) is nonsingular, the resulting iteration matrix equals
\[ hJ_2 := \begin{bmatrix} 1 - \hat{h}A & -\hat{h}B \\ C & O \end{bmatrix}. \] (6.25)

Performing LU-decomposition we find a similar expression as in (6.18), but with \( D = O. \) When the matrix \( CB \) is well-conditioned and bounded away from zero the inverse of \( hJ_2 \) is equal to \( (hJ_1)^{-1} (c + 0) \) (cf. (6.22)). This means that the numerical solution of the index one DAE (6.16) behaves like a solution of the associated index two DAE (6.24).

In Chapter 4 we have shown that this also holds for the exact solution. Rounding errors proportional to \( \eta \) and \( \eta h^{-1} \) are introduced in the variables \( x \) and \( y, \) respectively. This shows the ill-conditioning of a higher index DAE.
6.2. The Iteration Matrix

Example 6.5 Consider the following DAE of index two
\[ \dot{\mathbf{x}} = -\mathbf{y} + \mathbf{\sin t}, \]
\[ 0 = \mathbf{x} - \mathbf{\cos t}, \]
subject to the initial condition \( \mathbf{x}(0) = 1, \mathbf{y}(0) = 0 \). This DAE has solution \( \mathbf{x}(t) = \cos t, \mathbf{y}(t) = 2 \sin t \). Introducing artificial errors \( \delta \in [4 \cdot 10^{-3}, 6 \cdot 10^{-3}] \), into the system shows the influence of the rounding errors. On the interval \([0, 10^{-3}]\) we obtain from discretization with Euler backward the following table, where we use the following definitions: \( e_x := |x(t_n) - x_n|, \ e_y := |y(t_n) - y_n| \) and \( \text{drift} := |x_n - \cos t_n| \), i.e. the deviation from the constraint, for \( nh = T = 10^{-3} \). Table 6.2 shows that errors proportional to \( \delta \) appear in

<table>
<thead>
<tr>
<th>( h )</th>
<th>( e_x )</th>
<th>( e_y )</th>
<th>( \text{drift} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-4} )</td>
<td>0.54 \cdot 10^{-2}</td>
<td>0.57 \cdot 10^{-2}</td>
<td>0.54 \cdot 10^{-2}</td>
</tr>
<tr>
<td>( 10^{-5} )</td>
<td>0.60 \cdot 10^{-4}</td>
<td>0.62 \cdot 10^{-4}</td>
<td>0.60 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-6} )</td>
<td>0.51 \cdot 10^{-4}</td>
<td>0.44 \cdot 10^{-4}</td>
<td>0.51 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-7} )</td>
<td>0.59 \cdot 10^{-4}</td>
<td>0.72 \cdot 10^{-4}</td>
<td>0.59 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-8} )</td>
<td>0.54 \cdot 10^{-4}</td>
<td>0.20 \cdot 10^{-3}</td>
<td>0.54 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-9} )</td>
<td>0.58 \cdot 10^{-4}</td>
<td>0.11 \cdot 10^{-3}</td>
<td>0.58 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

Table 6.2. Results for the nearly index two index one DAE of Example 6.5 showing the \( h^{-1} \) effect in the algebraic variable \( y \). 

However, for \( \mathbf{CB} = \mathbf{rI} \) (see also Section 5.3), \( (\hat{\mathbf{C}}(\mathbf{I} - \hat{\mathbf{A}})^{-1}\mathbf{B})^{-1} \) has to be computed. For small fixed \( \hat{h} \) we find that
\[
(\hat{\mathbf{C}}(\mathbf{I} - \hat{\mathbf{A}})^{-1}\mathbf{B})^{-1} = \hat{h}^{-2}(\mathbf{CAB})^{-1} - (\hat{\mathbf{A}}\mathbf{B}(\mathbf{CAB})^{-1}) + O(\hat{h}) + O(\hat{h}^3), \quad (\hat{h} \to 0),
\]
when the matrix \( \mathbf{CAB} \) is well-conditioned and bounded away from zero. Hence,
\[
(\hat{\mathbf{A}}\mathbf{J})^{-1} = \begin{bmatrix}
-\hat{h}^{-1}\mathbf{B}(\mathbf{CAB})^{-1} & \hat{h}^{-2}\mathbf{B}(\mathbf{CAB})^{-1}
-\hat{h}^{-1}(\mathbf{CAB})^{-1} & \hat{h}^{-2}(\mathbf{CAB})^{-1}
\end{bmatrix} + \text{h.o.t.}, \quad (\hat{h} \to 0),
\]
implies that rounding errors proportional to \( n\hat{h}^{-1} \) and \( n\hat{h}^{-2} \) are introduced in the variables \( x \) and \( y \), respectively. Note that when \( \mathbf{CB} = \mathbf{O} \) in the DAE (6.24), discretizing this DAE results in the same iteration matrix (cf. (6.28)). This implies that, also numerically, the solution of the index two DAE is close to the solution of the associated index three DAE, i.e. \( \mathbf{CB} = \mathbf{O} \) in (6.24) (see also Chapter 5, where the analogue has been shown for the exact solution). If the matrix \( \mathbf{D} \) is ill-conditioned and \( \mathbf{CB} = \mathbf{O} \) in the DAE (6.16) then the inverse of the resulting iteration matrix \( \hat{\mathbf{A}}\mathbf{J} \) also results in (6.28) and the index one DAE (6.16) is numerically close to an index three DAE.
Next, consider the Hessenberg form DAE of index three
\[\begin{align*}
x' &= Ax + By + Cz + p, \\
y' &= Dx + Ey + q, \\
0 &= Fy + r,
\end{align*}\] (6.29)

where \( FDC \) is nonsingular. In this case the iteration matrix is
\[
hJ_1 := \begin{bmatrix}
1 - \hat{h}A & -\hat{h}B & -\hat{h}C \\
-\hat{h}D & 1 - \hat{h}E & 0 \\
0 & F & 0
\end{bmatrix}.
\] (6.30)

For a well-conditioned matrix \( FDC \), it has the inverse
\[
(hJ_1)^{-1} = \begin{bmatrix}
I - C(FDC)^{-1}FD & -h^{-1}C(FDC)^{-1}F & h^{-1}C(FDC)^{-1} \\
h(I - DC(FDC)^{-1}F)D & I - DC(FDC)^{-1}F & DC(FDC)^{-1} \\
-h^{-1}(FDC)^{-1}FD & -h^{-2}(FDC)^{-1}F & h^{-2}(FDC)^{-1}
\end{bmatrix} + \text{h.o.t.}
\] (6.31)

Therefore, rounding errors proportional to \( n \hat{h}^{-1} \), \( n \), and \( n \hat{h}^{-2} \) are introduced in the variables \( x \), \( y \), and \( z \), respectively, showing the ill-conditioning of a higher index DAE.

### 6.2.2 Effect of Rounding Errors on Solution Components

In Chapter 4, we shown that an index one DAE with an almost singular matrix \( D \) may effectively behave like a DAE of higher index. Consequently, we expect that the same may happen numerically. Therefore, problem (6.16) is considered once more. Now, we assume that the matrix \( D \) is almost singular, say \( D = \epsilon \tilde{D} \), where \( ||\tilde{D}^{-1}|| \approx 1 \), \( (\epsilon \to 0) \).

The inverse of the iteration matrix then differs from \((hJ_1)^{-1}\) as given by (6.20). In fact, one finds
\[
(hJ_1)^{-1} = \begin{bmatrix}
(I - \hat{h}A)^{-1}(I - \hat{h}BD^{-1}C(I - \hat{h}A)^{-1}) & \hat{h}(I - \hat{h}A)^{-1}BD^{-1} \\
-\tilde{D}^{-1}C(I - \hat{h}A)^{-1} \\
(I + \hat{h}A)(I - \hat{h}BD^{-1}C(I + \hat{h}A)^{-1}) & \hat{h}(I + \hat{h}A)BD^{-1}
\end{bmatrix} + \text{h.o.t.}
\] (6.32)

where
\[
\tilde{D} := D + \hat{h}C(I - \hat{h}A)^{-1}B.
\]

In order to quantify the behaviour of this matrix \((hJ_1)^{-1}\) more precisely, the inverse of the matrix \( \tilde{D} \) has to be studied into detail. For small \( \hat{h} \) it holds that
\[
\tilde{D} = D + \hat{h}C \sum_{i=0}^{\infty} \hat{h}^i A'B.
\] (6.33)
6.2. The Iteration Matrix

We can partition the matrix $D$ in a well- and an ill-conditioned part using again the GSVD of $D$, cf. (4.76), i.e.

$$D = XTV^T,$$  \hspace{1cm} (6.34)

where

$$T = \text{diag}(T_{11}, T_{22}) \in \mathbb{R}^{m \times m},$$

$$T_{11} = \text{diag}(t_1, \ldots, t_r) \in \mathbb{R}^{r \times r}, \quad t_1 \geq t_2 \geq \ldots \geq t_r > 0,$$

$$T_{22} = \text{diag}(t_{r+1}, \ldots, t_m) \in \mathbb{R}^{(m-r) \times (m-r)}, \quad t_{r+1} \geq t_{r+2} \geq \ldots \geq t_m > 0,$$

i.e. the matrix $D$ has numerical rank $r$. Hence, $D^{-1}$ can be written as

$$D^{-1} = V(T + \hat{h}X^TC \sum_{i=0}^{\infty} \hat{h}^i A'BV)^{-1}X^T.$$  \hspace{1cm} (6.35)

Define the matrix $\tilde{D}$ as follows

$$\tilde{D} := T + \hat{h}X^TC \sum_{i=0}^{\infty} \hat{h}^i A'BV$$

$$= T + \hat{h}K.$$  \hspace{1cm} (6.36)

Partition $K$ similarly to $T$, i.e.

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}.$$  \hspace{1cm} (6.37)

Forming $\tilde{D}^{-1}$ by LU-factorization shows the importance of the term

$$\tilde{K} := \{E + \hat{h}K_{22} - \hat{h}^2 K_{21}(T + \hat{h}K_{11})^{-1}K_{12}\}^{-1}.$$  \hspace{1cm} (6.38)

Now, we can distinguish between the following cases:

(i). Let the matrix $(X^TCB)_{22} := \tilde{CB}_{22}$ be well-conditioned. Then $\tilde{K} = \hat{h}^{-1}\tilde{CB}_{22}^{-1}$ + h.o.t., for $h > \varepsilon$. Effectively, this means that we can partition $y$ in index one and index two variables $z_1$ and $z_2$, respectively (see also Section 4.5), by defining

$$V^Ty := \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}.$$  \hspace{1cm} (6.39)

We conclude that $\|\tilde{K}\| = O(\min(\varepsilon^{-1}, \hat{h}^{-1}))$. If we have rounding errors $\delta$, say, this explains the appearance of errors proportional to $\delta \varepsilon^{-1}$ for $h < \varepsilon$, as we have seen in Table 6.1, cf. Example 6.4.

(ii). Let $\tilde{CB}_{22} = 0$, whereas $\tilde{CAB}_{22}$ is well-conditioned. In particular, this means that $\tilde{K} = \hat{h}^{-2}(\tilde{CAB}_{22} - \tilde{CB}_{22}T^{-1}\tilde{CB}_{12})^{-1}$ + h.o.t., for $\hat{h}^2 > \varepsilon$. In this case, the variable $z_2$ as defined in (6.39) is of index higher than two. Hence, $\|\tilde{K}\| = O(\min(\varepsilon^{-1}, \hat{h}^{-2}))$. 

(iii). Let the matrix $\bar{\mathbf{C}}\bar{\mathbf{B}}_{22}$ be singular. This means that variable $z_2$ consists not only of variables of index two, but also of variables of index three or higher. Now, $||\bar{\mathbf{K}}||$ is at least of order $\min(\epsilon^{-1}, \bar{h}^{-m+1})$.

(iv). The procedure above can be repeated. Let $m$ be the highest effective index of any variable, then

$$||\bar{\mathbf{K}}|| = O(\min(\epsilon^{-1}, \bar{h}^{-m+1})).$$

**Remark 6.6** More intuitively the previous analysis can be seen as follows. Consider $\mathbf{D} + \bar{h}C(I - \bar{h}A)^{-1}B$. For small $\bar{h}$ this matrix can be written as

$$\mathbf{D} + \bar{h}C(I - \bar{h}A)^{-1}B = \mathbf{D} + \bar{h}CB + \bar{h}^2CAB + \bar{h}^3CA^2B + \cdots.$$ 

Suppose that both $\mathbf{D}$ and its inverse have a norm of moderate size, then

$$(\mathbf{D} + \bar{h}C(I - \bar{h}A)^{-1}B)^{-1} = \mathbf{D}^{-1} + O(\bar{h}).$$

On the other hand if the matrix $\mathbf{D}$ is almost singular, say $\mathbf{D} = \epsilon\hat{\mathbf{D}}$, for simplicity, with $||\hat{\mathbf{D}}^{-1}|| \approx 1$, ($\epsilon \to 0$), and $||C\mathbf{B}|| \leq M$ for a constant $M$ of moderate size, then

$$(\mathbf{D} + \bar{h}C(I - \bar{h}A)^{-1}B)^{-1} \approx \bar{h}^{-1}((C\mathbf{B})^{-1}).$$

Now, it is immediately obvious that

$$||D + \bar{h}C(I - \bar{h}A)^{-1}B|| \approx O(\min(\epsilon^{-1}, \bar{h}^{-m+1})).$$

where $m$ is the highest effective index of any of the variables. So, the conditioning of the iteration matrix is practically determined by the highest effective index of any variable.

From the foregoing analysis we can conclude that the actual conditioning constant of the iteration matrix may behave like one associated with a DAE of a higher index than expected if any of the variables has a higher index effectively; this implies that errors proportional to $\epsilon\bar{h}^{m+1}$ are introduced into the solution, if $m$ denotes the effective index of the DAE; so care has to be taken to ensure that the effect of rounding errors remains small and to prevent these rounding errors from dominating the discretization error. In linear systems as considered in this section such $O(\bar{h}^{-m+1})$ errors are introduced in the algebraic variables only. However, for more general systems, like quasilinear or nonlinear systems, the state variables can also be strongly affected by such errors (see [59]), because of mixing of the errors. For such systems this means that the Newton iterations may fail to converge for smaller step-sizes, cf. Example 6.9 in Section 6.4. In the next section we develop a method to minimize the effect of rounding errors in the discretization.
6.2.3 Stabilized Index Reduction

In Chapter 3 we have shown that DAEs of higher index, i.e., index \( n \geq 2 \), depend on derivatives of the forcing functions up to order \( n-1 \). In the Subsections 6.2.1 and 6.2.2 we have shown that the iteration matrices, obtained by discretization of DAEs of higher index, are ill-conditioned and that rounding errors proportional to \( \eta \) and \( \eta h^{-1} \) appear in the state and the algebraic variables, respectively. Here, we study the effect of a stabilization method as introduced in Section 6.1. Consider again the following DAE of index two

\[
\begin{align*}
x' &= Ax + By + p, \\
0 &= Cx + q.
\end{align*}
\]

Now, we stabilize this DAE in the following manner

\[
\begin{align*}
x' &= Ax + By + L\mu + p, \\
0 &= Cx + q, \\
0 &= CAx + CBy + Cp + q',
\end{align*}
\]

where \( CL \) is a well-conditioned and therefore invertible matrix. It is obvious that the Lagrange multiplier \( \mu \) is an index two variable, whereas the algebraic variable \( y \) is an index one variable in this case. Discretization of the latter system results in the following iteration matrix

\[
\bar{h}_J_{\text{stab}} = \begin{bmatrix}
I - \hat{h}A & -\hat{h}B & -\hat{h}L \\
C & 0 & 0 \\
CA & CB & 0
\end{bmatrix}.
\]

Now, one finds for the inverse of this iteration matrix

\[
(\bar{h}_J_{\text{stab}})^{-1} = \begin{bmatrix}
I - Q & L(CL)^{-1} & \hat{h}(B(CB)^{-1} - L(CL)^{-1}) \\
(CB)^{-1}CA(Q-I) & -(CB)^{-1}CAL(CL)^{-1} & (CB)^{-1} \\
-\hat{h}^{-1}(CL)^{-1}C & \hat{h}^{-1}(CL)^{-1} & -\hat{h}(CL)^{-1}
\end{bmatrix} + h.o.t.,
\]

where the projector \( Q \) is defined by \( Q := L(CL)^{-1}C \). Clearly, \( (\bar{h}_J_{\text{stab}})^{-1} = \mathcal{O}(\hat{h}^{-1}) \) and similarly for the inverse of \( \bar{h}_J \). However, in the stabilized system (6.41) rounding errors proportional to \( \eta \hat{h}^{-1} \) are introduced in the Lagrange multipliers \( \mu \) only, whereas in the original index two DAE (6.40) such errors appear in the algebraic variable \( y \). Remember that this Lagrange multiplier satisfies \( \mu = 0 \), implying that the largest errors are introduced in the unimportant variables only.

In the previous subsection, we have shown that for index one DAEs (4.10) with an almost singular matrix \( D \) the conditioning of the iteration matrix \( h_J \) (16.17) is determined by the variables with the highest effective index. This suggests the following method for improving the conditioning of the linear system: Consider again the index one DAE (6.16). In addition, let \( D = \epsilon I \) and \( CB \) be well-conditioned. This means that
this index one DAE has index two effectively, i.e. rounding errors proportional to $\eta \hat{h}^{-1}$ are introduced into the algebraic variable $y$. First we put $D = 0$, thereby considering the associated reduced system, which is index two in this case. Secondly, the reduced DAE is stabilized as in (6.41), preventing the introduction of errors proportional to $\eta \hat{h}^{-1}$ in the $y$ variables. For more complicated matrices $D$ this method has the disadvantage that one has to distinguish between the variables which effectively behave as variables of index one and the variables which effectively are of higher index.

However, in the following we will show an interesting phenomenon: to prevent the introduction of errors proportional to $\eta \hat{h}^{-1}$ in the $y$ variable, it is not necessary to put $D$ to zero first and stabilizing the system afterwards. Instead it is possible to improve the conditioning by performing the stabilization at once (i.e. without putting $D = 0$), which is much easier in situations where the matrix $D$ is more complicated than $\eta I$, since we do not have to distinguish between variables which are effectively index one or index two then. To this end consider the following system

$$
\begin{align*}
x' &= Ax + By + L \mu + p, \\
0 &= Cx + Dy + q, \\
0 &= CAx + CB y + D y' + C p + q'.
\end{align*}
(6.42)
$$

After discretization of this problem the inverse of the iteration matrix

$$
\hat{h} J_{\text{stab}} = \begin{bmatrix}
I - \hat{h}A & -\hat{h}B & -\hat{h}L \\
C & D & 0 \\
\hat{h}CA & \hat{h}CB + D & 0
\end{bmatrix}
$$

reads

$$
(\hat{h} J_{\text{stab}})^{-1} = \begin{bmatrix}
I - Q & L(\text{CL})^{-1} & B(\text{CB})^{-1} - L(\text{CL})^{-1} \\
(\text{CB})^{-1}CA(Q - I) & -(\text{CB})^{-1}CAL(\text{CL})^{-1} & \hat{h}^{-1}(\text{CB})^{-1} \\
\hat{h}^{-1}(\text{CL})^{-1}C & \hat{h}^{-1}(\text{CL})^{-1} & \hat{h}^{-1}(\text{CL})^{-1}
\end{bmatrix}
$$

Note that the third block column of the above matrix is scaled by a factor $\hat{h}^{-1}$, which arises from differentiating $y$ in (6.42); this factor will not cause any problems, since the matrix $(\hat{h} J_{\text{stab}})^{-1}$ operates on a right-hand side vector with a factor $\hat{h}$ appearing in the corresponding third row. Hence, in the thus stabilized system errors proportional to $\eta$ will not appear in the approximation of the variables $x$ and $y$, whereas errors proportional to $\eta \hat{h}^{-1}$ will appear in the newly defined variable $\mu$. So, this stabilization method gives the same results concerning the conditioning as the aforementioned method.

If the matrix $D$ is well-conditioned, one can show that

$$
(\hat{h} J_{\text{stab}})^{-1} = \begin{bmatrix}
I - Q & L(\text{CL})^{-1} & -L(\text{CL})^{-1} \\
\hat{h}D^{-1}CA(Q - I) & -\hat{h}D^{-1}CAL(\text{CL})^{-1} & D^{-1} \\
\hat{h}^{-1}(\text{CL})^{-1}C & \hat{h}^{-1}(\text{CL})^{-1} & \hat{h}^{-1}(\text{CL})^{-1}
\end{bmatrix} + \text{h.o.t.,}
$$
6.2. The Iteration Matrix

and the conditioning of the DAE will be the same as before for the index one variables, whereas the index two variable \( \mu \) has a conditioning constant \( O(h^{-1}) \). The second row of \( [A_{m,2b}]^{-1} \) shows that errors proportional to \( \eta \dot{h} \) are introduced in well-conditioned algebraic variables. Combining these two results shows that, whenever the matrix \( D \) is ill-conditioned, one may stabilize without making any factorizations of \( D \) in order to determine the effective index of any of the variables. Hence, we have constructed a method to cheaply improve the conditioning of an index one DAE, which contains variables that are effectively of higher index.

Example 6.7 Consider an index one DAE (6.16), with

\[
A = \begin{bmatrix} -2 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & B_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix},
\]

and \( D = \begin{bmatrix} D_{11} & 0 \\ D_{22} & 0 \end{bmatrix} \)

where \( p_1(t) = q_1(t) = t/2 + \sin t, \quad p_2(t) = t + 2\cos t, \) and \( q_2 = \cos t - \sin t. \)

The reduced DAE (i.e. \( \varepsilon = 0 \)) has the solution

\[
x_1(t) = -q_2(t), \quad x_2(t) = x_2(0) \exp(-t),
\]

\[
y_1(t) = x_2(0) \exp(-t) - t/2 - \cos t, \quad \text{and} \quad y_2(t) = 2 \sin t.
\]

For \( \varepsilon \) approaching to zero, the algebraic variable \( y_2 \) behaves effectively like an index two variable, whereas \( y_1 \) is an index one variable. According to Subsection 6.2 rounding errors proportional to \( \hat{h}^{-1} \) are introduced in the numerical approximation of \( y \) due to the ill-conditioning of the iteration matrix. Let us choose the initial values \( x_1 = 2 \) and \( x_2 = 0 \).

Using the stabilized form (6.42) we obtain from discretization with Euler backward on the interval \( I = [0, 10^{-3}] \) the Table 6.3. Here \( e_{x_i} := |x_i(t_h) - x_{i,0}|, e_{y_i} := |y_i(t_h) - y_{i,0}|, \) \( i = (1, 2), \) and drift := \( |(Cx + Dy)_{n+1}|, \) i.e. the deviation from the constraint, for \( nh = T = 10^{-3} \).

<table>
<thead>
<tr>
<th>( h )</th>
<th>( e_{x_1} )</th>
<th>( e_{x_2} )</th>
<th>( e_{y_1} )</th>
<th>( e_{y_2} )</th>
<th>drift</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-4} )</td>
<td>0.55 \cdot 10^{-4}</td>
<td>0.99 \cdot 10^{-4}</td>
<td>0.39 \cdot 10^{-4}</td>
<td>0.12 \cdot 10^{-3}</td>
<td>0.55 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-3} )</td>
<td>0.60 \cdot 10^{-4}</td>
<td>0.11 \cdot 10^{-3}</td>
<td>0.50 \cdot 10^{-7}</td>
<td>0.13 \cdot 10^{-3}</td>
<td>0.60 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>0.49 \cdot 10^{-4}</td>
<td>0.90 \cdot 10^{-4}</td>
<td>0.54 \cdot 10^{-7}</td>
<td>0.12 \cdot 10^{-3}</td>
<td>0.49 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-3} )</td>
<td>0.56 \cdot 10^{-4}</td>
<td>0.11 \cdot 10^{-3}</td>
<td>0.55 \cdot 10^{-7}</td>
<td>0.11 \cdot 10^{-3}</td>
<td>0.56 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>0.58 \cdot 10^{-4}</td>
<td>0.11 \cdot 10^{-3}</td>
<td>0.55 \cdot 10^{-7}</td>
<td>0.11 \cdot 10^{-3}</td>
<td>0.58 \cdot 10^{-4}</td>
</tr>
<tr>
<td>( 10^{-3} )</td>
<td>0.58 \cdot 10^{-4}</td>
<td>0.12 \cdot 10^{-3}</td>
<td>0.55 \cdot 10^{-7}</td>
<td>0.11 \cdot 10^{-3}</td>
<td>0.59 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

Table 6.3 Results for the nearly index two index one DAE of Example 6.7 showing that the stabilization method can circumvent the \( \hat{h}^{-1} \) effect in the algebraic variable \( y_2 \).
6.3 ERROR ESTIMATION FOR STEP-SIZE CONTROL AND TERMINATION OF NEWTON ITERATIONS

This section explains how the step-size control of BDF codes, and in particular DASSL (cf. [66]), works. In Subsection 6.2.1 we have shown that the iteration matrix used by e.g. BDF is poorly conditioned when the step-size is small. We will show that this may cause various complications in the numerical solution by (variable step-size) BDF codes when solving DAEs of higher index. These problems can be circumvented by scaling the error estimates, i.e. both the estimates for the local error and the estimates for terminating the Newton iterations.

Here, we concentrate on the general (nonlinear) implicit form of a DAE, motivated by the fact that we use a code, i.e. DASSL (cf. [66]), which requires the system of equations in this implicit form. In order to understand these problems we first investigate the error equation for the numerical solution by a BDF code of the DAE (3.1). Application of a $k$-step BDF method leads to the following difference equation

$$f(t_n, x_n, h_n^{-1} \rho_n(E)x_{n-1}) = 0,$$  (6.45)

where, as introduced in Section 3.2, $\rho_n$ is the generating polynomial of the variable step-size BDF method, i.e. $\rho_n(z) := \sum_{k=0}^{k-1} a_k z^{k-1}$. Let $\bar{x_n}$ be the exact solution of this difference equation. The previous equation is solved by Newton iteration. This results in the numerical solution $x_n$ of (6.45), which introduces an error $e_n := \bar{x_n} - x_n$, due to terminating the Newton iterations early. Let $e_n := x_n - x(t_n)$ be the global error. Then we get

$$0 = f(t_n, \bar{x_n}, h_n^{-1} \rho_n(E)\bar{x}_{n-1})$$
$$= f(t_n, x_n + e_n, h_n^{-1} \rho_n(E)\bar{x}_{n-1} + e_{n-1})$$
$$= f(t_n, x_n + e_n, h_n^{-1} \rho_n(E)(x_n+e_n) + e_{n-1} + e_{n-2}).$$

The local truncation error $\tau_n$, say, is defined by $\tau_n := \rho_n(E)\bar{x}(t_n) - h_n x'(t_n)$. We find

$$0 = f(t_n, x_n + e_n, x_n, h_n^{-1} \tau_n + h_n^{-1} \rho_n(E)(e_{n-1} + e_n)).$$

Using Taylor expansions around $x(t_n)$, $x'(t_n)$ we find the linearization (cf. (4.1a))

$$0 \simeq A_n h_n \tau_n + h_n^{-1} \rho_n(E)(e_{n-1} + e_n)) + B_n (e_n + e_{n-1}),$$  (6.46)

where $A_n$ and $B_n$ are defined by

$$A_n := \frac{d}{dx} f(t_n, x(t_n), x'(t_n)) \quad \text{and} \quad B_n := \frac{d}{dx} f(t_n, x(t_n), x'(t_n)),$$

respectively. We thus find

$$e_n = -A_n^{-1}(\bar{A}_n + B_n B_n)^{-1} \bar{A}_n \tau_n + \sum_{i=1}^{k} a_i n e_{n-1} + \sum_{i=1}^{k} a_i n e_{n-1} - e_n.$$  (6.47)
where \( \tilde{h}_n := h_n / o_n \) (since \( \beta_0 = 1 \)) as in Subsection 6.2.1. In Subsection 6.2.1 we have shown that the iteration matrix \( hJ \) is poorly conditioned for DAEs of higher index. Note that the scaled matrix \( \left[ \begin{array}{c} h_{n-1} \\ \tilde{h}_n \end{array} \right] hJ \) is equal to the matrix \( \tilde{h}_n \tilde{A}_n + \tilde{h}_n \tilde{B}_n \). Together with the error equation (6.47) this implies that some of the variables can have large errors, since they are amplified by positive powers of \( \tilde{h}_n^{-1} \). This will cause failure of the error estimating process both with respect to the step-size control and with respect to terminating the Newton iterations; this then will result in inefficient behaviour or even failure of the code. To understand these effects we will examine the relation between the local error and the truncation error. In first order terms this relationship is given by

\[
\tilde{e}_n = -o_{n+1} (\tilde{A}_n + \tilde{h}_n \tilde{B}_n)^{-1} \tilde{A}_n \tau_n.
\]

From equation (6.20) we can obtain the following estimate for \( \tilde{e}_n \) as a function of \( \tau_n \) in the index one case

\[
\tilde{e}_n = \begin{bmatrix} O(1) & O(1) \\ O(1) & O(h^n) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tau_n = \begin{bmatrix} O(1) \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} \tau_n.
\]

hence the truncation error \( \tau_n \) is not amplified. Now, the step-size must be chosen such that the estimation \( \text{EST}_n \) of the local error \( \tilde{e}_n \) is less than a given tolerance. It is clear that for the index one DAE the same error estimation as for ODEs can be used, since for ODEs we use the relation \( \tilde{e} = \tau_n \approx O(\tilde{h}_n^{n+1}) \). However, for semi-explicit index two systems (6.24) we find with the help of (6.22) that

\[
\tilde{e}_n = \begin{bmatrix} O(1) & O(\tilde{h}_n^{-1}) \\ O(h^n) & O(h^n) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tau_n = \begin{bmatrix} O(1) \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} \tau_n,
\]

which implies that in this case the truncation error \( \tau_n \) is amplified by \( \tilde{h}_n^{-1} \) and that the local error will not be of order \( O(h_n^{n+1}) \) as expected by the code. In order to use a criterion for step-size control, we use the scaled error estimate

\[
\text{EST}_n = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{h}_n \end{bmatrix} \tilde{e}_n.
\]

which is \( O(h_n^{n+1}) \) as the code expects. Care has to be taken to ensure that the scaled error estimate \( \text{EST}_n \) is of order of the tolerance \( \text{TOL} \). Hence, we allow the largest errors in the variables of index two which is more or less satisfactory since in general these are the ill-posed variables. Index three Hessenberg systems (6.29) yield

\[
\tilde{e}_n = \begin{bmatrix} O(1) & O(\tilde{h}_n^{-1}) & O(h^n) \\ O(h^n) & O(1) & O(h^n) \\ O(\tilde{h}_n^{-1}) & O(h^n) & O(h^n) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tau_n = \begin{bmatrix} O(1) & O(h^n) \\ O(h^n) & O(1) \\ O(\tilde{h}_n^{-1}) & O(h^n) \end{bmatrix} \begin{bmatrix} 0 & 0 \end{bmatrix} \tau_n.
\]

This implies that we should use the scaled error estimate

\[
\text{EST}_n = \begin{bmatrix} O(h_n) & 0 & 0 \\ 0 & O(1) & 0 \\ 0 & 0 & O(\tilde{h}_n^{-1}) \end{bmatrix} \tilde{e}_n
\]
in order to get $\text{EST} \approx \text{TOL}$. Here, again, the larger errors are allowed in variables of higher index. For equations of motion for mechanical systems, this means that the position variables are controlled to order $\hat{h}^{k+1}$ while the velocities and the Lagrange multipliers are controlled to order $\hat{h}^k$ and $\hat{h}^{k-1}$, respectively.

The global error $e_n$ and the error $e_n$, due to terminating the Newton iterations after a finite number of steps, are approximately related by the following recurrence relation

$$e_n = -a_{0,n} (\hat{A}_n + \hat{h}_n \hat{B}_n)^{-1} \hat{A}_n (\sum_{i=1}^{k} a_i e_{n-i} + \sum_{i=1}^{k-1} a_{i+1} e_{n-1-i}) - e_n.$$ 

This shows that the local error is influenced not only by the error $e_n$ at time $t_n$, but also by the errors $e_{n-1}, \ldots, e_{n-k}$ made at previous times. This gives the following relationship between the local error $\hat{e}$ and the error $\epsilon$

$$\hat{e}_n = -e_n - a_{0,n} (\hat{A}_n + \hat{h}_n \hat{B}_n)^{-1} \hat{A}_n \sum_{i=1}^{k} a_i e_{n-i}.$$

(6.49)

Hence, $\hat{e}$ and $\epsilon$ are related by the matrix factor $(\hat{A}_n + \hat{h}_n \hat{B}_n)^{-1} \hat{A}_n$. Again, this means that some components of $\epsilon$ are amplified by positive powers of $\hat{h}_n^{-1}$. Therefore, some variables have to be solved more precisely than others to limit the influence of the algebraic errors on the local error in order to let the local error $\hat{e}$ be of the order of the tolerance (i.e. $\hat{e} \approx \mathcal{O}(\text{TOL})$). We can see from (6.49) that the amplification of the $i$-th component of $\epsilon$ is found by considering the maximum power of $\hat{h}_n^{-1}$ in the $i$-th column of $(\hat{A}_n + \hat{h}_n \hat{B}_n)^{-1} \hat{A}_n$.

To this end we have to scale the estimates for terminating the Newton iteration in such a way that the scaled error estimates are of the order of the tolerance. Consider for example the semi-explicit index two DAE (6.24). From (6.48) we can see that the $x$-components of $\epsilon$ (say $e_x$) are not amplified at all, whereas the $y$-components of $\epsilon$ (i.e. $\epsilon_y$) are amplified by $\hat{h}_n^{-1}$. In order to let the local error $\hat{e}$ be of the order of the tolerance we must require for the estimates of the local error that $\text{EST}^x_n \approx \text{TOL}$ and $\hat{h}_n \text{EST}^y_n \approx \text{TOL}$. Hence, the scaling vector is $[1, \hat{h}_n^{1/2}]$. For the Heisenberg index three system (6.29) we see that $e_x$ is amplified by $\hat{h}_n^{-1}$ and $e_y$ is amplified by $\hat{h}_n^{1/2}$, whereas $e_z$ is not amplified at all. This implies that we must choose the scaling vector $[\hat{h}_n, \hat{h}_n^{3/2}, 1]^{T}$ to obtain $\hat{e} = \mathcal{O}(\text{TOL})$.

In the foregoing analysis we have studied the propagation of errors both caused by the truncation error and caused by terminating the Newton iterations early. We showed that for DAEs of higher index some components of the errors may be amplified by powers of $\hat{h}_n^{-1}$, which leads to severe complications for the step-size control and the termination of the iterations. This may result in failure of DAE codes designed for DAEs of index one (e.g. DASSL). However, with the suggested modifications the usual error estimates can be applied. That means that DAEs of higher index can be solved by DAE codes designed for DAEs of lower index. Indeed, solving higher index DAEs with the modified code is comparable to solving index one DAEs (see [8]). A disadvantage of this method is that now some variables are controlled less accurately, which suggests that those variables may exhibit larger errors than expected.
6.4 APPLICATION TO MULTIBODY SYSTEMS

With the modifications of the previous section we can use the BDF code DASSL to solve DAEs of higher index efficiently. Therefore, we can solve the equations of motion for multibody systems in index three form (3.19). However, since the velocity variables are scaled by $\dot{\lambda}$ and the algebraic variables are scaled by $\lambda^2$, the solution of the index three DAE can exhibit large errors. Therefore, we reduce the index and use the projection method (6.12) in order to alleviate drift. Now, only $\lambda$ has to be scaled by $\hat{\lambda}$. In the following example we compare both methods for the crank-slider mechanism shown in Figure 1.5 (see also Example 2.6).

Example 6.8 Let the lengths of the rods be $l_1 = 1$, $l_2 = 2$ and the mass be $m_1 = 1$, $m_2 = 2$. With initial values $\phi_1(0) = 0.5236$ and $\phi_2(0) = 0$, we find the solution as plotted in Figure 6.4 for $0 \leq t \leq 10$. The plot of the $y$ coordinates of the bodies $B_1$ and $B_2$ shows that $y_1 = y_2$, which is caused by constraining the distal end of rod $B_2$ to move along the $x$-axis. In Table 6.4 we compare the numerical solution of the index three DAE and the projection method. The error $\epsilon$, say, at the endpoint of the integration interval in the numerical solution of the DAE (3.1) is the weighted root mean square norm (cf. [9]).
defined by
\[ e := \frac{1}{n} \left( \sum_{i=1}^{n} \left( \frac{w_i - w_i'}{w_i} \right)^2 \right)^{1/2} \text{TOL}, \]
where
\[ w_i = |x_i|^2 |RTOL + ATOL|, \]
for chosen tolerances \( TOL, ATOL \) and \( RTOL \). Here and in the sequel we will take \( TOL = ATOL = RTOL \). As the exact solution \( \mathbf{x}^e \) we have used the numerical solution computed with an extremely tight error tolerance \( TOL = 10^{-30} \). Furthermore, \( nste, nfe \) and \( njf \) denote the number of steps, the number of function evaluations and the number of Jacobian evaluations, respectively. From Table 6.4 we conclude that the numerical solution of the projection method is computed much more efficiently than the solution of the index three problem.

Let us consider the double pendulum shown in Figure 1.4 and considered in the Examples 2.4 and 2.5.

**Example 6.9** For this double pendulum we formulate the equations of motion in descriptor form (cf. Subsection 2.1.3). Therefore, we choose as generalized coordinates \( \mathbf{q} = [q_1, q_2]^T \), where the vectors \( q_1 = [\dot{x}_1, y_1, \dot{\phi}_1]^T \) and \( q_2 = [\dot{x}_2, y_2, \dot{\phi}_2]^T \) (with the angle \( \alpha = \phi_1 - \phi_2 - 2\pi \)) denote the Cartesian vectors of the generalized coordinates for \( B_1 \) and \( B_2 \), respectively. The equations of motion for the double pendulum constitute of the dynamic equations

\[
\begin{align*}
    m_1 \ddot{x}_1 &= m_1 g - \lambda_1, \\
    m_1 \ddot{y}_1 &= -\lambda_2, \\
    J_1 \ddot{\phi}_1 &= (\lambda_2 + 2\lambda_3) I_1 \cos \phi_1 - (\lambda_1 + 2\lambda_3) I_1 \sin \phi_1, \\
    m_2 \ddot{x}_2 &= m_2 g - \lambda_3, \\
    m_2 \ddot{y}_2 &= -\lambda_4, \\
    J_2 \ddot{\phi}_2 &= \lambda_4 I_2 \cos \alpha - \lambda_3 l_2 \sin \alpha. 
\end{align*}
\]

(6.51a)
6.4. Application To Multibody Systems

and the position constraints

\[ 0 = \phi = [\phi_1, \phi_2], \]

(6.51b)

where

\[ \phi_1(q) = \begin{bmatrix} x_1 - l_1 \cos \varphi_1 \\ y_1 - l_1 \sin \varphi_1 \end{bmatrix} \quad \text{and} \quad \phi_2(q) = \begin{bmatrix} x_2 - 2l_1 \cos \varphi_1 - l_2 \cos \alpha \\ y_2 - 2l_1 \sin \varphi_1 - l_2 \sin \alpha \end{bmatrix}. \]

When all eigenvalues of the generalized mass matrix \( M \) (cf. Subsection 2.1.2) are comparable, i.e. \( m_1 \) and \( m_2 \) are similarly, numerically solving the system both in the index three form (6.51) and in the projection form (6.12) gives similar results. However, in case of a heterogeneous multibody system, i.e. a system which includes bodies of very different masses, these methods behave differently. Let \( m_1 \gg m_2 \). This implies that the matrix \( \phi_1 M^{-1} \phi_1^T \) becomes almost singular, since this matrix has two eigenvalues of order \( m_1/m_2 \). Such a system (6.51) is comparable to the index one and two DAEs with almost singular matrices \( D \) and \( CB \) considered in the Chapters 4 and 5, respectively. However, there is no associated higher index (i.e. index four) system available. Consequently, this DAE is ill-conditioned for \( m_1 \gg m_2 \). Letting \( m_1 \) become larger with regard to \( m_2 \) shows that DASSL fails in the index three formulation (6.51), whereas DASSL applied to the projection form (6.12) still works well (even for larger differences between \( m_1 \) and \( m_2 \)). This is caused by the fact that in the projection form the conditioning is governed by the inverse of the matrix \( \phi_1 \phi_1^T \), which is well-conditioned. This projection formulation, therefore, results in a well-conditioned DAE.

Table 6.5 shows a comparison of the numerical solutions of both the index three system (6.51) and the projected system (6.12) for \( l_1 = 1, l_2 = 2, m_1 = 10^9 \) and \( m_2 = 2 \cdot 10^{-4} \). The angles at initial time were chosen as \( \varphi_1(0) = \alpha(0) = \pi/2 \), while the initial velocities were taken equal to zero. In Table 6.5 \( \epsilon_{p_1} \) and \( \epsilon_a \) denote the global errors in the angles \( \varphi_1 \) and \( \alpha \), respectively, where we have used the angles found by integrating the state space form with a tolerance of \( 10^{-10} \) as the exact solution. For the tolerances where \( n/s/t/e \) is denoted by an * the computations crash. Hence, Table 6.5 clearly illustrates that the projected form (6.12) is to be preferred to the index three formulation (6.51). The solution of this double pendulum is plotted in Figure 6.5.

<table>
<thead>
<tr>
<th>Tol</th>
<th>(6.51)</th>
<th>(6.12)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n/s/t/e )</td>
<td>( \epsilon_{p_1} )</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>493</td>
<td>0.33 \cdot 10^{-1}</td>
</tr>
<tr>
<td>( 10^{-5} )</td>
<td>730</td>
<td>0.14 \cdot 10^{-2}</td>
</tr>
<tr>
<td>( 10^{-6} )</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>( 10^{-8} )</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>( 10^{-10} )</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 6.5 Comparison of the numerical solution of the double pendulum in index three form (6.51) and in projected form (6.12).
Figure 6.5  Solution for the double pendulum for $TOL = 10^{-6}$.

Hence, solving the equations of motion for multibody systems in the projection formulation (6.12) with the BDF code DASSL using the appropriate modifications (see Subsection 6.3) results in an efficient and robust method for simulating multibody dynamics. The use of the aforementioned projection form results in a larger system than the original index three form. One may conclude that as a result, the linear algebra might be much more expensive than in the original form. Fortunately, the efficiency of the method can be improved further by reducing the amount of work in the linear algebra as suggested in [34, 43]. Each step of a BDF method requires the solution of linear systems (cf. (6.46)) of the form

$$
\begin{bmatrix}
-\hat{h}^{-1} & 1 & O & \phi_\delta^T \\
O & -\hat{h}^{-1} & M & \phi_q^T \\
\phi_q & O & O & O \\
O & \phi_q & O & O
\end{bmatrix}
\begin{bmatrix}
\delta q \\
\delta v \\
\delta \lambda \\
\delta \mu
\end{bmatrix}
= 
\begin{bmatrix}
a \\
b \\
c \\
d
\end{bmatrix}
$$

(6.52)

found by supplying an analytical Jacobian, neglecting the derivatives of $M$, $f$ and $\phi_q$.

Solving (6.52) is equivalent to solving two smaller linear systems, viz.

$$
\begin{bmatrix}
-\hat{h}^{-1} & O \\
\phi_q & O
\end{bmatrix}
\begin{bmatrix}
\delta v \\
\delta \lambda
\end{bmatrix}
= 
\begin{bmatrix}
b \\
d
\end{bmatrix}
$$

and

$$
\begin{bmatrix}
-\hat{h}^{-1} & \phi_q^T \\
\phi_q & O
\end{bmatrix}
\begin{bmatrix}
\delta q \\
\delta \mu
\end{bmatrix}
= 
\begin{bmatrix}
a - \delta v \\
c
\end{bmatrix}
$$

(6.53)

Implementing this may lead to a considerable reduction in computing time (in [43] a factor three is reported for some problems).
7

DISCONTINUITIES IN MECHANICAL SYSTEMS

Many systems in simulation and control give rise to systems of differential algebraic equations (DAEs) with forcing functions containing discontinuities in the form of finite jumps either in components of the forcing functions themselves or in some derivatives of them. Quite often multi-body systems exist in an environment where impulsive forces are exerted on the system. These impulsive forces are defined as large forces exerted for a short time interval. Examples of systems experiencing impulsive forces are models of crash-victims during a crash, machine couplers and robot grippers. Force discontinuities also occur in conversion from slip to stick conditions and vice versa, which can happen in the presence of Coulomb friction. Other kinds of singularities that often occur for mechanical systems are e.g. contact phenomena, hysteresis and functions supplied by data bases.

The lack of smoothness causes numerical difficulties, since the convergence theory of discretization schemes as well as the step-size and order control usually require differentiability up to some given order. As a consequence, the step-size selection may break down and there is no safe local error estimation anymore. Therefore, a numerical integrator will behave very inefficiently in the presence of a singularity.

In the following, the occurring difficulties are explained and demonstrated by numerical examples. A method is developed to handle the discontinuities in an efficient and robust way.

7.1 NUMERICAL PROBLEMS DUE TO SHOCKS AND DISCONTINUITIES

In many applications, the appearance of discontinuities can be modelled by the use of switching functions, resulting in discontinuous differential algebraic equations of the fol-
lowing form

\[ \begin{align*}
    \dot{x} &= \begin{cases} 
        f_1(t, x, y), & \text{if } s(t, x) > 0, \\
        f_2(t, x, y), & \text{if } s(t, x) < 0,
    \end{cases} \\
    0 &= g(t, x, y),
\end{align*} \tag{7.1a} \tag{7.1b} \]

where \( s \) is the switching function.

Using these switching functions the singularity is localized at the root of \( s \). We shall call the surface \( S := \{(t, x) \mid s(t, x) = 0\} \) the manifold of discontinuity. Furthermore, we define the regions \( S^+ := \{(t, x) \mid s(t, x) > 0\} \) and \( S^- := \{(t, x) \mid s(t, x) < 0\} \). Inside these regions \( S^+ \) and \( S^- \) the dynamic equation (7.1a) is characterized by the forcing functions \( f_1 \) and \( f_2 \), respectively. In either region, the problem (7.1) can be solved by classical methods.

A discontinuous differential equation can be integrated numerically in various ways. A first method is to use a variable step, variable order integration method with local error control, thereby hoping that the step-size mechanism will handle the discontinuity appropriately. In [36] and [42] examples are given, which show how multistep and Runge-Kutta methods deal with such a singularity. Both methods detect the discontinuity fairly well, albeit by means of numerous rejected steps, since the presence of a discontinuity is usually noticed by a very large value of the local error estimate; the latter fact results in rejecting the step and a dramatic reduction of the step-size and possibly of the order of the method. Back again in the smooth region the step-size will usually be increased until the code tries to step over the singularity again. Then the same procedure is repeated and the code will pass the singularity only after several repetitions of this procedure. So, in general many steps may be rejected when passing such a discontinuity, yielding a very inefficient method. This is illustrated by the following.

**Example 7.1** Consider the ODE

\[ \dot{x} = \begin{cases} 
    0, & t \leq 10, \\
    5, & t > 10,
\end{cases} \quad x(0) = 10. \tag{7.2} \]

Figure 7.1 shows the inefficient behaviour of a BDF code (with \( TOL = 10^{-4} \)) when ignoring the discontinuity. \[ \square \]

However, it might even be possible that the integrator misses the discontinuity completely. Consider e.g. the following.

**Example 7.2**

\[ \begin{align*}
    \dot{x} &= \begin{cases} 
        5, & t \leq 10, \\
        10, & 10 < t < 11, \\
        5, & t \geq 11,
    \end{cases} \quad x(0) = 0.
\end{align*} \tag{7.3} \]
7.2. TREATMENT OF DISCONTINUITIES

BDF fails for this rather simple system because it steps over the discontinuity and will follow the solution branch \( x = 5t \). This effect is due to so-called superstability of BDF (cf. [55, 60]) and it will occur for all tolerances. □

So, standard codes for the numerical solution of ODEs handle, if at all, discontinuities inefficiently. Therefore, integration codes have been modified to detect singularities more precisely and to handle them more appropriately. These singularity detecting codes have been developed for both multistep methods (cf. [36]) and Runge-Kutta methods (cf. [24]). They are based on estimates of the order and magnitude of the discontinuity and hence the step-size which keeps the error under control when passing a discontinuity.

Despite the efforts spent to construct such singularity detecting codes, the best available methods to handle the appearance of discontinuities are methods which use switching functions to explicitly indicate the appearance of a discontinuity. These methods stop the integration at the surface of the discontinuity and restart the computation with the new right-hand side. These methods appear to be not only more efficient but also more accurate in integrating discontinuous problems.

7.2 TREATMENT OF DISCONTINUITIES

In the previous Chapter 6 we have shown that the problem (7.1) can be solved by classical methods in either of the regions \( S^- \) and \( S^+ \). Problems occur when the switching function
s changes sign. To understand this phenomenon, let us consider the scalar products

\[ a_1 := \frac{\partial \phi}{\partial x} f_1 + \frac{\partial \phi}{\partial \tau}, \quad (\tau, x) \in S^+, \]
\[ a_2 := \frac{\partial \phi}{\partial x} f_2 + \frac{\partial \phi}{\partial \tau}, \quad (\tau, x) \in S^-, \]  

which denote the total derivatives of the switching function in the direction of the solution \( x \) determined by the right-hand side functions \( f_1 \) and \( f_2 \), respectively. Existence of classical solutions of (7.1) beyond the manifold of discontinuity depends on the sign of \( a_1 \) and \( a_2 \). If \( a_1 > 0 \), this means that the flow of (7.1) goes from \( S^- \) towards \( S^+ \), whereas it goes into the direction of \( S^- \) when \( a_1 < 0 \). For \( a_2 \) and the flow for \( f_2 \) similar arguments show the same sort of behaviour. In the degenerate case that either \( a_1 = 0 \) or \( a_2 = 0 \) the flow is locally tangent to \( S \). As a result, one can distinguish between the following four cases considering the vector fields near \( S \), shown geometrically in Figure 7.2:

(i). \( a_1 > 0, a_2 > 0 \): the flow crosses \( S \) from \( S^- \) to \( S^+ \);
(ii). \( a_1 > 0, a_2 < 0 \): on both sides of \( S \) the flow moves away from \( S \);
(iii). \( a_1 < 0, a_2 < 0 \): the flow crosses \( S \) from \( S^+ \) to \( S^- \);
(iv). \( a_1 < 0, a_2 > 0 \): on both sides of \( S \) the flow moves towards \( S \).

![Figure 7.2](image)

Figure 7.2: Vector fields near the discontinuity.

Both in case (i) and (iii), the classical solution can be continued across \( S \). However, in case (ii) the solution is not unique and bifurcations may occur. In case (iv), the problem
7.3. Occurrence of Discontinuities in Multibody Dynamics

has no longer a classical solution since the solution is trapped in the discontinuity manifold $S$. However, one can show that the system has a generalized solution (cf. [25]), then, which says that the generalized solution has its derivative in the convex hull of $f_1$ and $f_2$, i.e.

$$ x = v f_1 + (1 - v) f_2, \quad 0 \leq v \leq 1. $$

The value $v$ acts as a control variable which forces the solutions to stay in the manifold $S$ (as proposed in [19] and [27]), implying that the solution has to satisfy the DAE

$$
\begin{align*}
\dot{x} &= v f_1 + (1 - v) f_2 =: \bar{f}(t, x, y, v), \\
0 &= g(t, x, y), \\
0 &= s(t, x).
\end{align*}
$$

(7.5a)  
(7.5b)  
(7.5c)

It is obvious that the control variable $v$ is an index two variable, since

$$
\frac{\partial g}{\partial \bar{f}} = a_1 - a_2 \neq 0.
$$

This implies that one can solve for $v$ from (7.5a) and (7.5c), indeed, taking the total derivative of (7.5c) and substituting (7.5a) yields

$$
sv_1 f_1 + (1 - v)s_1 f_2 + s_t = 0,
$$

implying that

$$

v = - \frac{s_1 + s_t f_2}{s_1 f_1 - a_2 f_2} = - \frac{a_2}{a_2 - a_1}. \quad (7.6)
$$

Substitution of this value of $v$ into (7.5a) results in the DAE

$$
\begin{align*}
\dot{x} &= \frac{a_2}{a_2 - a_1} f_1 - \frac{a_1}{a_1 - a_2} f_2, \\
0 &= g(t, x, y).
\end{align*}
$$

(7.7a)  
(7.7b)

Hence, the solution in the manifold $S$ is described by the vector field which satisfies the DAE (7.7). The solution of the discontinuous DAE (7.1) will leave the discontinuity manifold $S$ when $v = 0$ or $v = 1$, meaning that one of the scalars $a_1$ or $a_2$ changes sign.

7.3 OCCURRENCE OF DISCONTINUITIES IN MULTIBODY DYNAMICS

In Section 7.1 we have explained the numerical problems in dealing with discontinuities. Treatment of these shocks by using switching functions means that all possible switching functions must be provided by the user, it is convenient to have at one’s disposal some sort of library with possible switching functions and methods to deal with them. For that reason, it is important to recognize the various kinds of discontinuities which may occur when modelling mechanical systems. The possibly appearing discontinuities can be subdivided into several classes, e.g.
(i). *Impacts.* These impacts arise naturally in the crash-safety field, where one studies the response of multibody systems during crashes. They are characterized by instantaneous appearing (and possibly very large) forces. These forces cause discontinuities in the velocities, since the movement of the mechanical system may be stopped abruptly.

**Example 7.3** Consider for example a pendulum moving under the influence of gravity, cf. Example 2.1. After some time this pendulum hits a wall (cf. Figure 7.3). On impact the velocity of the pendulum will almost immediately drop to zero due to a suddenly increasing elasto-plastic force as a function of the penetration into the wall. The example will be worked out in Example 7.9 and numerical results shown in Figure 7.7.

![Figure 7.3](image)

\[ Y \]

**Figure 7.3** Impact of pendulum into a wall.

(ii). *Contact problems* for rigid bodies give rise to algebraic equations as the touching of bodies leads to algebraic constraints in a natural way.

(iii). *Coulomb friction.* When one solid body slides over another or over a surface, dry friction or Coulomb friction is generated. Like other frictional forces, this force is dissipative and nonlinear.

**Example 7.4** Consider a solid block sliding with a velocity \( \dot{q} \) relative to a rough solid plane (cf. [19, 39]). The force of the plane acting on the block consists of

![Figure 7.4](image)

\[ q \]

\[ \dot{q} \]

\[ \mu N \]

\[ \mu N \]

\[ \mu N \]

**Figure 7.4** Coulomb friction.
the normal force $N$ perpendicular to the surface and the Coulomb friction force $F_f$ parallel to the surface and opposing the motion. The classical law of sliding friction states that this force can be expressed as

$$F_f = -\mu N \text{sgn}(v^T q) e,$$  \hfill (7.8)

where $e$ is the unit vector tangential to the surface. The Coulomb friction coefficient $\mu$ depends on the roughness of the surfaces and the materials used. The component in the positive $e$-direction of the friction force, i.e., $F_f = e^T F_f$, is plotted against the velocity in Figure 7.4 for the case of a constant normal force. In the so-called stick phase, i.e., when $e^T q = 0$, the friction force is not uniquely determined. Then, the method, which has been explained in Section 7.2 (for $a_1 < 0$, $a_2 > 0$), may be applied, since $|F_f| \leq \mu |N|$, always.

(iv). 

**Hysteresis** describes the energy dissipation which can occur in mechanical systems where several bodies can interact with each other, and in elasto-plastic materials. Hence, hysteresis is e.g. very important in the simulation of the interaction between the airbag and the steering wheel and in the modelling of seat belts. Consider the problem of Example 7.3 again:

**Example 7.5** Due to the elasto-plastic force the pendulum will start moving into the opposite direction after some time. Because of energy dissipation the unloading elasto-plastic force will follow another curve than the loading force. This phenomenon is called hysteresis.

(v).

**Input functions given in tabular form**, to define e.g. material properties, acceleration fields, spring forces as a function of the relative elongation, and so on.

(vi).

**Breaking of interconnections between bodies**: The removal of a joint can lead to a rapid movement of the remaining system and to a sudden increase of the constraint forces in one or more of the remaining constraints (see [56]).

### 7.4 USAGE OF THE SWITCHING FUNCTIONS

The appearance of discontinuities is detected using switching functions as is already explained in Section 7.2. In this section we discuss the numerical simulation of systems exhibiting discontinuities. In Section 7.2 it was shown that a sign change of the switching function indicates the passing of a discontinuity. In order to simulate this phenomenon effectively one has to adjust the numerical integration method in order to incorporate the use of switching functions. This implies that the numerical integrator has to be extended with the following method, which follows from the analysis of Section 7.2 for treating discontinuous systems like (7.1) (cf. also [19, 42, 83]):

(i). Let $x_0$ be the initial value at initial time $t_0$. If $s(t_0, x_0) = 0$ go to step (ii). Otherwise, if $s(t_0, x_0) \neq 0$ then solve the dynamic equation $\dot{x} = f$, together with the constraint
\[ 0 = g(t, x) \] as long as \((t_0, x_0) \in \mathbb{R}^+\). However, when \((t_0, x_0) \in \mathbb{R}^-\) one has to solve the differential equation \(\dot{x} = f_1(t, x)\) together with the constraint equation \((7.1)\) until \((t, x) \in \mathbb{R}\). When \((t, x) \in \mathbb{R}\) go to step (ii).

(iii) Compute \(a_1\) and \(a_2\) as defined by (7.4).

(iv) If \(a_1 > 0\) and \(a_2 > 0\), then solve the DAE (7.1) with \(\dot{x} = f_2(t, x)\) until \((t, x) \in \mathbb{R}\). Go to step (ii).

(v) If \(a_1 < 0\) and \(a_2 < 0\), then solve the DAE (7.1) with \(\dot{x} = f_2(t, x)\) until \((t, x) \in \mathbb{R}\). Go to step (ii).

After the localization of a root, i.e. \(s(t_*, x_*) = 0\), the forcing function is changed to the new mode and the discretization is adapted to the new situation. This means that the code should e.g. be restarted for one-step methods or that the discretization order should be reduced. However, it appears that it is most effective to restart also in the case of multistep methods, since otherwise information of previous steps has to be used. Unfortunately, this information does not have to be related in any way to the new situation which occurs after the discontinuity.

### 7.4.1 Root Finding

We have shown in Section 7.1 that it is important to find the location of the switch. The switch can e.g. be localized using an algorithm for detecting a root of the switching function, cf. DASSLRT (cf. [9]). In the sequel we proceed with a switching function denoted by a vector function \(s\). When a sign of any component of the switching function is changed, the code locates the root of \(s\) by a root search using an interval method like the Illinois algorithm (cf. [17]). This method is a modification of the secant method and gives cubic convergence with three function evaluations. The root finding mechanism will be explained briefly.

After a successful step from \(t_0\) to \(t_{i+1}\), \(s(t_{i+1}, x_{i+1})\) is evaluated. If \(s(t_0, x_0)\) and \(s(t_{i+1}, x_{i+1})\) have a different sign in any of the components, a switch must have occurred in \([t_i, t_{i+1}]\). Then the root solver is used to find this switch, i.e. a root of \(f(z)\) in the interval \([z_0, z_1]\) is sought, using secant-like iterations. Hence,

\[
z_2 = z_1 - \frac{f_1}{f_1 - f_0} (z_1 - z_0). \tag{7.9}
\]

After evaluating \(f(z)\) the next secant step is performed until convergence. In case that \(f\) is a vector function, say \(f \in \mathbb{R}^n\), we want to find the leftmost root of \(f\) in \([z_0, z_1]\). Therefore, we determine the maximum quotient \(-\frac{f_1}{f_1 - f_0}\) and then

\[
z_2 = z_1 - \frac{f_{\text{max},1}}{f_{\text{max},1} - f_{\text{max},0}} (z_1 - z_0). \tag{7.10}
\]
7.4. Usage of the Switching Functions

For the evaluation of \( s(t, x(t)) \) interpolation is used. Disadvantages of this method are:

(i). The root finder may fail due to ill-conditioned switching functions with a large uncertainty domain.

(ii). The aforementioned method might experience problems with more-dimensional nonlinear switching functions.

The following example illustrates the first problem.

**Example 7.6** Consider the ODE

\[
\dot{x} = \begin{cases} 
   f_1 := 3(t-3)^2, & 2 - x > 0, \\
   f_2 := \sin(t-3), & 2 - x < 0,
\end{cases} \tag{7.11}
\]

subject to \( x(0) = -25 \). Clearly the solution reads

\[
x(t) = \begin{cases} 
   2 + (t-3)^3, & 2 - x > 0, \\
   3 - \cos(t-3), & 2 - x < 0.
\end{cases} \tag{7.12}
\]

The switching function is given by \( s(x) = 2 - x \) and the switching parameters are given by \( a_1 = -3(t-3)^2 \) and \( a_2 = -\sin(t-3) \), respectively. In the exact solution the switch takes place at \( t = 3 \) with \( x(3) = 2 \). Starting in \( S^+ \) the solution switches to \( S^- \). However, due to numerical errors the numerical switch may be situated to the left of \( t = 3 \). For \( t = 3 - \delta \) \((0 \leq \delta \leq 1)\) the system is like the case \( a_1 < 0, a_2 > 0 \); this results in a numerical solution which will stick on \( S^+ \) during some time until the solution moves into \( S^- \).

Table 7.1 shows some characteristics of the numerical solution when integrating from \( t = 0 \) to \( t = 4 \). The error \( e \) at \( t = 4 \) is defined by \( e := x(4) - x_0 \), where \( x_0 \) is the approximation of \( x(4) \). For tolerances smaller than \( 10^{-4} \) the numerical integration fails.

<table>
<thead>
<tr>
<th>Tol</th>
<th># steps</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^{-2} )</td>
<td>45</td>
<td>( 0.13 \cdot 10^{-1} )</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>56</td>
<td>( 0.65 \cdot 10^{-1} )</td>
</tr>
</tbody>
</table>

Table 7.1 Comparison of BDF methods in order to show the effect of the use of switching functions.

So, a method based on root finding will fail when integrating discontinuous differential equations like the one shown in the aforementioned Example 7.6. In the next section, we will develop a more robust method, which is able to solve such equations as will be shown in Example 7.8.
7.4.2 An Algorithm Based on Monitoring The Switching Functions

One can easily give alternative ways of using switching functions. For instance we can use the switching functions to monitor the sign of \( s \). The method works as follows. If the switching function changes sign, the step-size \( h \) and possibly the order of the discretization are reduced until a solution is reached for which no sign change occurs, or which lies close to the roots within some prescribed accuracy. In the latter case the solution is close to the discontinuity and in order to accurately integrate the solution past the discontinuity, the integration is continued performing a restart. This can be explained in a more detailed way by considering Figure 7.5 which visualizes the possible function values of the switching function. The whole region is divided into four intervals depending on the value of the monitor function, i.e. \( I_1 = (-\infty, -\delta), I_2 = (-\delta, 0), I_3 = (0, \delta) \) and \( I_4 = (\delta, \infty) \). Here, \( \delta \) is of the order of the integration tolerance. Together, the intervals \( I_2 \) and \( I_3 \) form a small layer of size \( 2\delta \) around the root of the switching function. Let \( x_\alpha \) be the numerical solution on \( t = t_\alpha \) and define \( \alpha := s(t_\alpha, x_\alpha) \). At the next integration step from \( t_\alpha \) to \( t_{\alpha+1} = t_\alpha + h_\alpha \) we find \( x_{\alpha+1} \) and we can evaluate \( s_{\alpha+1} \). However, when \( s \) switches from \( I_1 \) to \( I_2 \) or back, the discontinuity is skipped over meaning that \( x_{\alpha+1} \) may become a completely wrong approximation of \( x(t_\alpha) \). Therefore, such a transition is forbidden and \( x_{\alpha+1} \) is not accepted. This means that the step-size \( h_\alpha \) will have to be reduced until the numerical integrator accepts a solution \( x_{\alpha+1} \) that lies in the same interval as \( x_\alpha \) or until \( x_{\alpha+1} \in I_2 \) or \( x_{\alpha+1} \in I_3 \). When \( s \) switches from \( I_1 \) (or possibly \( I_3 \)) to \( I_2 \) or \( I_3 \), this implies that the solution lies in the small layer around the discontinuity and the discontinuity is localized and the discretization is restarted from \( t = t_{\alpha+1} \).

Example 7.7 To show the effect of using switching functions in the numerical simulation of discontinuous systems, consider Example 7.2 once more. Table 7.2 shows the number of steps for several tolerances, for the various methods. For this particular problem we conclude from Table 7.2 that a standard BDF method not adapted to discontinuities (denoted as "ignoring" in Table 7.2) is not useful. In addition, the same BDF method combined with either the rootfinding method or the monitoring method performs satisfactorily, both yielding comparable to the number of steps and resulting errors.

Let us consider Example 7.6 once more.

Example 7.8 The discontinuous ODE (7.12) of Example 7.6 is now integrated using a BDF method combined with the monitoring algorithm. The results are summarized in Table 7.3, showing that this method handles the discontinuity well, contrary to the same BDF method combined with rootfinding (cf. Table 7.1).
7.5. USE OF THE METHOD IN MECHANICAL PROBLEMS

<table>
<thead>
<tr>
<th>Tol</th>
<th># steps</th>
<th>e</th>
<th># steps</th>
<th>e</th>
<th># steps</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>15</td>
<td>0.50 · 10^1</td>
<td>39</td>
<td>0.71 · 10^0</td>
<td>39</td>
<td>0.11 · 10^0</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>20</td>
<td>0.50 · 10^1</td>
<td>46</td>
<td>0.81 · 10^{-2}</td>
<td>58</td>
<td>0.81 · 10^{-2}</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>23</td>
<td>0.50 · 10^1</td>
<td>68</td>
<td>0.42 · 10^{-4}</td>
<td>82</td>
<td>0.14 · 10^{-3}</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>26</td>
<td>0.50 · 10^1</td>
<td>93</td>
<td>0.16 · 10^{-6}</td>
<td>103</td>
<td>0.11 · 10^{-6}</td>
</tr>
<tr>
<td>$10^{-10}$</td>
<td>30</td>
<td>0.50 · 10^1</td>
<td>111</td>
<td>0.83 · 10^{-9}</td>
<td>111</td>
<td>0.29 · 10^{-8}</td>
</tr>
</tbody>
</table>

Table 7.2 Comparison of BDF methods in order to show the effect of using switching functions.

<table>
<thead>
<tr>
<th>Tol</th>
<th># steps</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>40</td>
<td>0.20 · 10^{-1}</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>64</td>
<td>0.14 · 10^{-2}</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>98</td>
<td>0.47 · 10^{-6}</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>130</td>
<td>0.10 · 10^{-8}</td>
</tr>
</tbody>
</table>

Table 7.3 Comparison of BDF methods in order to show the effect of the use of switching functions.

Hence, the monitoring method is more robust than root finding, while the efficiency is comparable.

7.5 USE OF THE METHOD IN MECHANICAL PROBLEMS

In this section we consider mechanical systems subject to discontinuities. In the Examples 7.3 and 7.5 the problem of a moving pendulum hitting a wall is described. This mechanical system exhibits discontinuities due to the impact and to the phenomenon of hysteresis. This system is considered again in the following example.

Example 7.9 The equations of motion for the pendulum of Figure 7.3 constitute of the dynamic equations

$$
\begin{align*}
\ddot{x} &= mg - \lambda_1, \\
\ddot{y} &= -\lambda_2, \\
J\ddot{\phi} &= -\lambda_1 l \sin \phi + \lambda_2 l \cos \phi, \quad J = \frac{1}{2}ml^2
\end{align*}
$$

(7.13a)

together with the constraint equations

$$
\begin{align*}
x - l \cos \phi &= 0, \\
y - l \sin \phi &= 0,
\end{align*}
$$

(7.13b)
as long as gravity is the only externally applied force on the pendulum (see Example 2.1 of Subsection 2.1.1, for the dimensions). The system is subject to the initial conditions \( \varphi(0) = \varphi_0 \) and \( \psi(0) = 0 \). At some angle, say \( \varphi = \varphi_\ast \), the free end of the pendulum hits the wall and an elasto-plastic penetration force \( F_{\text{el}} \) is generated during the penetration of the pendulum into the wall. This elasto-plastic force is a function of the penetration. The force will follow a hysteresis curve (cf. Figure 7.6), since the loading and the unloading of the force take place along different curves due to the fact that considerable energy is lost in this process due to plastic deformations. Figure 7.6 shows that the dynamic equations are subject to four discontinuities, i.e. when the penetration \( p \) equals \( 0, p_1, p_{\text{max}} \) or \( p_f \). The effects of the impact and hysteresis are taken into account using the following mechanical model:

\[
\begin{align*}
mx &= mg + \lambda_1, \\
m\dot{y} &= -\lambda_2 + F_{\text{el}}, \\
J\ddot{\psi} &= -\lambda_3 \sin \varphi + \lambda_4 \cos \varphi + F_{\text{el}} l \cos \varphi. 
\end{align*}
\]  

(7.14)

The elasto-plastic penetration force \( F_{\text{el}} = F_{\text{el}}(p)e_y \), where \( p \) denotes the penetration, i.e.

\[
p = p(\varphi) := \begin{cases} 0, & \text{if } \varphi > \varphi_\ast, \\ 2l \sin(\varphi_\ast - \varphi), & \text{if } \varphi \leq \varphi_\ast. \end{cases}
\]

(7.15)

is defined according to Figure 7.6 by

\[
F_{\text{el}} := \begin{cases} 0, & \text{if } p = 0, \\ c_1 p, & \text{if } 0 < p \leq p_1 \land (\dot{p} > 0 \lor p_{\text{max}} \leq p_1), \\ c_1 p_1 + c_2 (p - p_1), & \text{if } p_1 < p \leq p_{\text{max}} \land \dot{p} > 0, \\ c_3 p + c_4, & \text{if } p_f < p < p_{\text{max}}, \dot{p} < 0. \end{cases}
\]

(7.16)
where for $p \leq p_f$ the unloading penetration force $F_{ul}$ becomes zero. The constants $c_1$, $c_2$ and $c_3$ are material constants and $c_4$ depends on the penetration $p_{max}$. We find $c_4$ from

$$c_4 p_{max} + c_4 = c_1 (p_{max} - p_f) = c_1 p_f + c_3 (p_{max} - p_f),$$

hence

$$c_4 = -c_1 p_f = (c_1 - c_2) p_f + (c_2 - c_3) p_{max}.$$

Let us choose $f = 0.9$, $m = 30$ and $g = 9.81$, and the initial conditions $\varphi(0) = 1.85$ and $\dot{\varphi}(0) = 0$. For the impact we take $\varphi_u = 0$. Moreover, the elasto-plastic force is defined by $c_1 = 8 \cdot 10^5$, $c_2 = 1 \cdot 10^6$, $c_3 = 2 \cdot 10^6$ and $p_f = 5 \cdot 10^{-3}$. In Figure 7.7 the numerical solution of this system is plotted, for the $x$, as well as for the $y$, the $\varphi$ and the $\dot{\varphi}$-variable. It shows that on impact, i.e. at $\varphi = \varphi_u$, the velocity of the pendulum drops almost at once to zero due to the increasing penetration force. Then $\varphi$ reaches its minimum and the velocity $\dot{\varphi}$ even becomes slightly negative which means that the pendulum starts moving into the opposite direction under influence of this penetration force. However, the penetration force is unloading and becomes equal to zero (cf. also Figure 7.6). This implies that the pendulum is going to oscillate until its motion damps out due to the loss of energy.

Figure 7.7 Numerical solution of crashing pendulum computed with BDF code with $TOL = 10^{-4}$ using the monitoring algorithm.
The next example handles a mechanical system that consists of a weight subject to Coulomb friction (cf. [42, 44, 70, 78]).

**Example 7.10** Let us consider a body $B$ which slides in a non-smooth groove filled with a viscous medium as shown in Figure 7.8. The body is attached to a spring. Let $m$ be the mass of the body, $k$ the spring constant, $c$ the damping coefficient and $W$ the maximum Coulomb friction force. Suppose that the body is forced to move by an externally applied periodical force $F$. The system is described by the dynamic equation

$$m\ddot{x} + c\dot{x} + W\text{sgn}(\dot{x}) + kx = F(\Omega t).$$

(7.17)

By defining

$$\omega = \sqrt{k/m}, \quad \tau = \omega t, \quad 2D = c/\sqrt{mk}, \quad 0 < D < 1$$

$$\mu = W/k, \quad \eta = \Omega/\omega, \quad \frac{1}{k}F(\Omega t) = F(\eta t).$$

equation (7.17) can be expressed in dimensionless form as

$$x'' + 2Dx' + \mu \text{sgn}(x') + x = F(\eta t).$$

(7.18)

where $x' = \frac{dx}{d\tau}$. Equation (7.18) shows that the velocity $x'$ is the switching function. The switches $a_1$ and $a_2$ on the manifold of discontinuity $S = \{t, x, x'|x' = 0\}$ are given by

$$a_1 = -x + P(\eta t) - \mu,$$

$$a_2 = -x + P(\eta t) + \mu.$$

In the case that $a_1 < 0$, $a_2 > 0$, the system is in the stick phase and is described by equation (7.5), where $x' = 0$ behaves like a constraint. The following values are taken for the parameters in (7.18): $D = 0.1, \mu = 4$, while the external force is $P(\eta t) = 2 \cos(\eta t)$.

The numerical solution of this mass-spring system is plotted in Figure 7.9. Starting with $x(0) = 3$ and $x'(0) = 4$, the solution $x$ reaches its maximum at $t = 0.563$. The body starts moving into the opposite direction until the system goes into the stick phase at $t = 2.04$.

During the stick phase the system is described by the DAE (7.5) and the control variable $\nu$ plays an important role. As a result of the externally applied force $P$ (or $F$), the system switches to the slip phase at $t = 2.63$, when $\nu$ reaches the value one. In the sequel, there appear several more transitions from the slip phase into the stick phase and vice versa. \(\square\)
Finally, we consider the crank-slider mechanism shown in Figure 1.5 in a slightly modified form.

Example 7.11 Here, we introduce Coulomb friction (with Coulomb friction coefficient $\mu$) for the pivot end of rod $B_2$ moving along the $x$-axis. In addition, let this pivot end be attached to $O$ by a spring (with spring constant $k$), the movement of the pivot end be damped by a damping force (with damping coefficient $c$). Furthermore, assume that a periodical force $F = F_\text{e}$ is applied to the same pivot end and that it is the only external force applied to this crank-slider mechanism. The equations of motion for this system during the slip phase are given by (here $\varphi_2$ differs from $\varphi_2$ in Figure 1.5, in so far that
here $\varphi_2$ is equal to $\varphi_1 + \varphi_2$ as chosen in Figure 1.5:

$$
\begin{align*}
    m_1 \dddot{x}_1 &= -\lambda_1, \\
    m_1 \dddot{y}_1 &= -\lambda_2, \\
    J_1 \dddot{\psi}_1 &= (\lambda_2 + 2\lambda_3) l_1 \cos \varphi_1 - (\lambda_1 + 2\lambda_3) l_1 \sin \varphi_1, \\
    m_2 \dddot{x}_2 &= -k_x x_p - cx_x - F(t) - \mu[N \text{sgn}(x_p)] - \lambda_3, \\
    m_2 \dddot{y}_2 &= -\lambda_4 - \lambda_5, \\
    J_2 \dddot{\psi}_2 &= (k_x + c_x + F(t) + \mu[N \text{sgn}(x_p)]) l_2 \sin \varphi_2 + (\lambda_4 + \lambda_5) l_2 \cos \varphi_2 - \lambda_3 l_2 \sin \varphi_2, \\
    0 &= x_1 - l_1 \cos \varphi_1, \\
    0 &= y_1 - l_1 \sin \varphi_1, \\
    0 &= x_2 - 2l_1 \cos \varphi_1 - l_2 \cos \varphi_2, \\
    0 &= y_2 - 2l_1 \sin \varphi_1 - l_2 \sin \varphi_2, \\
    0 &= y_p = y_2 + l_2 \sin \varphi_2.
\end{align*}

(7.19)

where the position vector of the distal end of pivot $B_2$ is denoted by $x_p$. Note that the normal force $N$ can be expressed in terms of $\lambda_5$, i.e., $N = -\lambda_5 e_y$. In more compact form this can be rewritten as

$$
\begin{align*}
    M \ddot{q} &= f - \mu[N \text{sgn}(e^T q)e - \Phi^T q], \\
    \theta &= \phi(q),
\end{align*}

(7.20)

where the vector $e$ is defined by $e = e(q) := [0, 0, 0, 0, 0, -1, \ldots, -1, m_2 e_2]^T$. So, the switching function $s$ is given by

$$
    s(q, q) = e^T q.

(7.21)

We obtain the scalar products $a_1$ and $a_2$, cf. (7.4), as

$$
\begin{align*}
    a_1 &= e^T M^{-1} (f - \mu[N e - \Phi^T q]) + \frac{\partial e}{\partial q} q, \\
    a_2 &= e^T M^{-1} (f + \mu[N e - \Phi^T q]) + \frac{\partial e}{\partial q} q.
\end{align*}

(7.22)

Let $a_1 < 0$ and $a_2 > 0$, i.e.,

$$
    e^T M^{-1} (f - \Phi^T q) + \frac{\partial e}{\partial q} q < \mu e^T M^{-1} e.

(7.23)

Then, the mechanism is in the stick phase, i.e., $e^T q = 0$, and the equations of motion can be written in the form of (7.5), viz.

$$
\begin{align*}
    M \ddot{q} &= f + (1 - 2\mu)[N e - \Phi^T q] = f + \lambda_i e - \Phi^T q, \\
    0 &= \phi(q), \\
    0 &= e^T \dot{q}.
\end{align*}

(7.24)
as long as $0 < \nu < 1$ or $|\lambda_0| < \mu|N|$. Note that the Lagrange multiplier $\lambda$, accounts for the constraint force resulting from the nonholonomic constraint $0 = e^T \dot{q}_i$, cf. (2.6). The control variable follows from (7.6), i.e.

$$
\nu = \frac{e^T M^{-1} (f + \mu |N| e - \phi^T \lambda) + q^T e \dot{q}_e}{2\mu |N| e^T M^{-1} e},
$$

implying that the mechanical system (7.24) can also be written as

$$
M \ddot{q} = f - \frac{e^T M^{-1} (f - \phi^T \lambda) + q^T e \frac{\partial e}{\partial q} e - \phi^T \lambda}{e^T M^{-1} e}.
$$

This holds as long as the system is in the stick phase (i.e. $e^T \dot{q}_i = 0$), hence until the inequality (7.23) is violated, i.e.

$$
|e^T M^{-1} (f - \phi^T \lambda) + q^T e \frac{\partial e}{\partial q} e| \geq \mu |N| e^T M^{-1} e,
$$

or

$$
|\lambda_0| \geq \mu |N|.
$$

since the system will go into the slip phase when the applied forces exceed the friction force. Let us choose $l_1 = 10$, $l_2 = 20$, $m_1 = m_2 = 1$, $\mu = 20$, $\kappa = 1$, $\epsilon = 5.9$, $F = 70 \cos(\pi t)$ and the initial values $q_1(0) = 3$ and $\phi_1(0) = 10^{-3}$. The numerical solution obtained for $TOL = 10^{-6}$ is plotted in Figure 7.10 then. Figure 7.10 illustrates that $x_p$ reduces several times to zero, implying the appearance of discontinuities. Starting in $S^+$ the crank-slider mechanism sticks for the first time at $t = 0.843$. As a result of the externally applied force $F$ the mechanism goes into the slip phase (i.e. $S^-$) at $t = 1.86$. At $t = 3.35$ the mechanism switches to $S^+$ and it sticks again at $t = 4.15$. Thereafter, the motion of the crank-slider mechanism has been damped out so much that other transmissions to the various phases can not be distinguished with the resolution of Figure 7.10 and the mechanical system approaches the stable state of equilibrium $q_1 = \pi$ and $\phi_2 = 0$. Moreover, one sees that $y_1 = y_2$, which is caused by the sliding along of the distal end of $B_1$ along the $x$-axis (i.e. $y_p = 0$).
Figure 7.10  Numerical solution of the crank-slider mechanism of Figure 7.8 under influence of Coulomb friction. The solution has been computed with a BDF code with $TOL = 10^{-5}$ using the monitoring algorithm.
8

CONCLUSIONS AND DISCUSSION

8.1 CONCLUSIONS

As stated in Section 1.4 the direct reason for the investigation described in this thesis was to develop a reliable and efficient numerical method for solving the equations of motion of multibody systems, in particular for systems with closed loops. Multibody dynamics often give rise to differential equations with forcing functions exhibiting discontinuities in the form of finite jumps, either in the function themselves or in some derivatives of them. As a consequence, the solution method should adapt to these discontinuities in order to simulate multibody systems exhibiting discontinuities in an efficient and robust manner.

The equations of motion describing the dynamics of multibody systems can be derived in two different ways. The first method is the so-called augmentation method. It uses a global coordinate description and adds the constraint equations (describing the interconnections between bodies) to the dynamic equations (i.e. differential equations), resulting in a system of differential algebraic equations (DAEs). Using this method the equations of motion are obtained in the descriptor form. The second method is the so-called elimination approach. It uses relative (independent) coordinates. For open loop systems (where the bodies are connected in a unique way) both the constraint forces and the constraint equations can be eliminated. The equations of motion then form a system of ordinary differential equations (ODEs). Closed loop systems (where the bodies are not connected in a unique way), however, also result in a system of DAEs using the elimination method, since additional constraints have to be satisfied.

Quite often in practice, DAEs appear in a form where the algebraic equations and the differential equations are explicitly given. Such systems are so-called semi-explicit DAEs. The DAEs describing the multibody dynamics are such semi-explicit systems. The nature of DAEs is described by the index, which appears to characterize the numerical difficulty of a DAE. ODEs are DAEs of index zero, and the equations of motion of
multibody systems are DAEs of index three.

In this thesis we have studied the conditioning, i.e. the sensitivity of the solution to perturbations in the system, of semi-explicit DAEs of index one and two. The matrices $D$, for index one DAEs (cf. Chapter 4), and $CB$, for index two DAEs (cf. Chapter 5), naturally appear when determining the dependency of the algebraic variable as function of the differential variables. Conditioning results in literature suggest that some conditioning constants depend on the norm of the inverse of these matrix coefficients, implying that these constants grow unboundedly when these matrices become almost singular. This means that small perturbations in the initial values or the forcing functions may result in large errors in the solution. In the Chapters 4 and 5 we have shown that such DAEs may behave almost like associated DAEs of higher index. Under the assumption that these associated higher index DAEs are well-conditioned themselves, appropriate stability constants of the latter are of moderate size. We have proven in this study that the conditioning constants of the index one and index two DAEs with almost singular matrices $D$ and $CB$, respectively, also remain moderately sized there. As a consequence, no larger errors will appear in the solution. By this we are able to determine the effective index of the variables and we give a method to determine, at least theoretically, the effective index of the variables. Moreover, it has been shown that the conditioning constants will grow unboundedly for DAEs that are not close to a well-conditioned higher index DAE. In addition we have shown for DAEs of index two and higher that perturbations of the coefficients may have a dramatic impact on the solution behaviour, whereas for DAEs of index zero and one this phenomenon does not occur.

As stated earlier, the index of a DAE characterizes the numerical difficulty of it. Most numerical methods are convergent for index one systems only and a loss of the accuracy may occur in the algebraic variables (i.e. those variables for which no differential equation is given directly). Methods based on backward differentiation formulae, so-called BDF methods, converge for some important classes of DAEs of index higher than one. The equations of motion for multibody systems, which have index three, belong to such a class. Applying BDF methods to these higher index DAEs results in ill-conditioned iteration matrices. The conditioning of iteration matrices has been studied in Section 6.2. This ill-conditioning implies that rounding errors and errors due to linearizations are amplified by negative powers of the step-size $h$ (see Section 6.2) in the numerical approximation of the algebraic variables. For more general DAEs, however, mixing of errors from the dynamic and the algebraic part can occur, implying that the dynamic variables are affected also. Hence, larger errors can appear in the approximations. We have shown that, for lower index DAEs which are almost higher index, this amplification effect will be worse than expected on ground of that lower index. Numerical evidence illustrates that the amplification of rounding errors in the numerical approximation commensurates with the theoretical conditioning results obtained in Subsection 6.2.2. For variable step, variable order BDF methods this ill-conditioning results in an erratic behaviour of the step-size control mechanism and the Newton iterations. As a consequence, these BDF methods might even fail integrating DAEs of index one. As suggested in literature, this can be remedied by scaling of the error estimates, both in the local error estimation and in the estimation of errors in the Newton iterations. Using these modifications means
that DAEs of higher index can be solved numerically in this higher index form directly. A particular projection method (6.12) combined with these modifications of the BDF method result in a rather efficient method for solving DAEs of higher index.

Discontinuities, e.g. due to impacts, hysteresis and Coulomb friction, arise frequently in the simulation of multibody systems. The appearance of such discontinuities causes severe problems during the numerical solution process; numerical solution methods may become very inefficient or may even fail during the passage of discontinuities. We have implemented a method to handle such discontinuities in an efficient way by using switching functions (see also [19, 43]). The use of these functions provides us a means to localize the discontinuity. Next we determine how the solution behaves near the discontinuity. When the solution passes the so-called manifold of discontinuity, the solution moves beyond the discontinuity and a new forcing function has to be considered. On the other hand, if the solution stays in the manifold of discontinuity (which may happen when a mechanical system with Coulomb friction switches from the slip phase to the stick phase for example), the DAE has to be augmented by constraining equations as well as constraint forces in order to keep the system in the manifold of discontinuity. Numerical examples studied in Chapter 7 show that this provides for an appropriate tool for the simulation of mechanical systems exhibiting discontinuities.

8.2 DISCUSSION

A subject for further development is the reduction of the amount of work in the linear algebra, which can be attained by exploring on the structure of the equations of motion for multibody systems. Especially in larger systems one can explore the sparseness of the system of linear equations when formulating the equations in descriptor form (cf. Chapter 2).

The numerical results for the solution of the equations of motion of multibody systems presented in Chapters 6 and 7 have been obtained by application of the descriptor form (cf. Chapter 2). To solve multibody systems with closed loops using the \( O(n) \)-recursive formalism for the generation of the equations of motion as elaborated in Chapter 2 has appeared to be suitable for large-scale open loop systems. Still, a subject for further study is the use of this formalism for closed loop systems.

The simulation illustrated in Figure 1.2 of Chapter 1 suggests the importance of handling multibody systems consisting not only of rigid bodies but also of flexible bodies. Induced by the increasing interest in deformable bodies, the development of \( O(n) \)-recursive methods (cf. Chapter 2) overlaps with the research activities on the modelling and the simulation of flexible bodies, cf. [16, 51–53]. Almost all approaches hitherto are based on the major assumption that the elastic deformations are small. One may use any multibody system formalism to generate the equations of motion. These equations are generated in the same form as for rigid bodies, but extra unknowns in the form of modal coordinates, accounting for the modal modes, have to be introduced into the system. In [51, 52] the \( O(n) \)-recursive method for generating the equations of multibody systems consisting of rigid bodies has been extended to a treatment of flexible bodies;
here the deformation modal coordinates are defined as generalized coordinates, i.e. the mass of the body is concentrated on certain nodes with elasticity remaining continuous. As a consequence, the resulting mechanical systems consisting of flexible bodies are more stiff than the systems constituted of rigid bodies only. Hence, a topic for further research is the study of the effects of incorporating flexibility on the numerical solution of the equations of motion of multibody systems.
REDUCTION OF EQUATIONS OF MOTION FOR A CHAIN

In this appendix we describe the procedure to reduce the equations of motion for the chain $\gamma$ of Figure 2.5 from tree end body $B_n$ to junction body $B_n$. Substitution of (2.27) and (2.25) into (2.34) gives

$$\sum_{i=1}^{n-1} \delta u_i^T [M_i \dot{v}_i - g_i] + (\delta u_{i-1}^T A_j^T + \delta q^T B_i^T) (M_n (A_n \dot{v}_{n-1} + B_n \dot{q}_n + c_n) - g_n)$$  \hspace{1cm} (A.1)

for virtual displacements $\delta u_i$ and $\delta q_n$. The kinematic constraints between bodies $B_{n-1}$ and $B_n$ may not be violated by $\delta q_n$. Since the relative joint coordinates $\dot{q}_n$ automatically comply with the constraints between bodies $B_{n-1}$ and $B_n$, $\delta q_n$ is kinematically admissible by definition. Therefore, the coefficients of $\delta q_n$ in equation (A.1) are zero. As a result, we obtain

$$\ddot{q}_n = -(B_n^T M_n B_n)^{-1} B_n^T [M_n (A_n \dot{v}_{n-1} + c_n) - g_n]. \hspace{1cm} (A.2)$$

where the existence of $(B_n^T M_n B_n)^{-1}$ can be proved according to [5]. Substitution of equation (A.2) into (A.1) gives

$$\sum_{i=1}^{n-2} \delta u_i^T [M_i \dot{v}_i - g_i] + \delta u_{i-1}^T ([M_{n-1} + \tilde{M}_{n-1}] \dot{v}_{n-1} - (g_{n-1} + \ddot{q}_{n-1})) = 0, \hspace{1cm} (A.3)$$

where the $\delta u_i$ are kinematically admissible for joints $i + 1, i + 2, \ldots, n - 1$.

$$\tilde{M}_{n-1} := A_j^T (I - M_n B_n) (B_n^T M_n B_n)^{-1} B_n^T M_n A_n \quad \text{and} \quad \ddot{q}_{n-1} := -A_j^T (I - M_n B_n) (B_n^T M_n B_n)^{-1} B_n^T [M_n c_n - g_n].$$

This reduction process can be continued to an arbitrary body $B_{n+1}$. By induction it can be shown that the following variational form of the equations of motion is obtained

$$\sum_{i=1}^n \delta u_i^T [M_i \dot{v}_i - g_i] + \delta u_{n+1}^T ([M_{n+1} + \tilde{M}_{n+1}] \dot{v}_{n+1} - (g_{n+1} + \ddot{q}_{n+1})) = 0. \hspace{1cm} (A.4)$$
where virtual displacements \( \delta u_i, i = 1, 2, \ldots, n \), are kinematically admissible for joints between bodies \( B_i \) to \( R_{n+1} \). However, they are free from kinematical admissibility conditions corresponding to constraints between bodies \( R_{n+1} \) to \( R_n \), because these constraints are satisfied by the relative coordinates.

Substituting equations (2.25) and (2.27), with \( i = m \) and \( j = m + 1 \), into (A.4) results in

\[
\sum_{i=1}^{m-1} \delta u_i^T [M_i \dot{v}_i - g_i] + \delta u_m^T [A_m^T [(M_{m+1} + \tilde{M}_{m+1})] [A_{m+1}] \dot{v}_m + \dot{M}_{m+1} q_{m+1} \\
+ c_{m+1} - (g_{m+1} + \tilde{g}_{m+1})] + M_m \ddot{v}_m - g_m + \delta q_{m+1}^T B_{m+1}^T [(M_{m+1} + \tilde{M}_{m+1})] A_{m+1} \dot{v}_m \\
+ \tilde{M}_{m+1} [(A_m^T \dot{v}_m + \dot{M}_{m+1} q_{m+1} + c_{m+1}) - (g_{m+1} + \tilde{g}_{m+1})] = 0,
\]

(A.5)

for kinematically admissible \( \delta u_i, i = 1, \ldots, m \), and \( \delta q_{m+1} \). Once more, \( \delta q_{m+1} \) is kinematically admissible by definition. As \( \delta u_i, i = 1, \ldots, m \), and \( \delta q_{m+1} \) are independent, the coefficient of \( \delta q_{m+1} \) in equation (A.5) must be equal to zero, resulting in

\[
\dot{q}_{m+1} = -(B_{m+1}^T [(M_{m+1} + \tilde{M}_{m+1})] B_{m+1})^{-1} B_{m+1}^T [(M_{m+1} + \tilde{M}_{m+1})] (A_{m+1} \dot{v}_m \\
+ c_{m+1} - (g_{m+1} + \tilde{g}_{m+1})).
\]

(A.6)

As before, according to [5], the existence of

\[
(B_{m+1}^T [(M_{m+1} + \tilde{M}_{m+1})] B_{m+1})^{-1}
\]

can be proved. Substituting (A.6) back into (A.5) gives

\[
\sum_{i=1}^{m-1} \delta u_i^T [M_i \dot{v}_i - g_i] + \delta u_m^T [(M_m + \tilde{M}_m) \ddot{v}_m - (g_m + \tilde{g}_m)] = 0,
\]

where

\[
\tilde{M}_m := A_m^T (I - (M_{m+1} + \tilde{M}_{m+1}) B_{m+1} [(B_{m+1}^T [(M_{m+1} + \tilde{M}_{m+1})] B_{m+1})^{-1} B_{m+1}^T] \\
\cdot (M_{m+1} + \tilde{M}_{m+1}) A_{m+1}
\]

(A.7)

and

\[
\tilde{g}_m := -A_m^T (I - (M_{m+1} + \tilde{M}_{m+1}) B_{m+1} [(B_{m+1}^T [(M_{m+1} + \tilde{M}_{m+1})] B_{m+1})^{-1} B_{m+1}^T] \\
\cdot ((M_{m+1} + \tilde{M}_{m+1}) c_{m+1} - (g_{m+1} + \tilde{g}_{m+1})),
\]

(A.8)

and \( \delta u_i \) is kinematically admissible for joints between bodies \( B_i \) to \( R_m \). This reduction process can be continued to junction body \( B_i \). Now, the reduced variational equations of motion read

\[
\delta u_i^T [(M_i + \tilde{M}_i^T) \dot{v}_i - (g_i + \tilde{g}_i^T)] = 0,
\]

where \( \delta u_i \) is kinematically admissible for all external constraints that act on junction body \( B_i \), and \( \tilde{M}_i^T \) and \( \tilde{g}_i^T \) follow from equations (A.7) and (A.8), respectively, with \( m = i \).
REDUCTION OF EQUATIONS OF MOTION FOR A CLOSED LOOP SYSTEM

We explain the reduction of the equations of motion for a closed loop system to a junction body. Substituting the kinematical relations, relating acceleration and virtual displacement of body $B_n$ in terms of body $B_{n-1}$ and the relative joint coordinates $q_n$ between them, into equation (2.41) we obtain

$$
\sum_{i=1}^{n-1} \delta u_i^T (M_i \ddot{q}_i - g_i) + (\delta u_i^T A_i + \delta q_i^T B_i^T) (M_n (A_n \ddot{q}_n - B_n \ddot{q}_n + c_n) - g_n + \phi_n^T \lambda) \\
+ \delta u_{n+1}^T [M_{n+1} \ddot{q}_{n+1} - g_{n+1} + \phi_{n+1}^T \lambda] + \sum_{i=m+2}^{m} \delta u_i^T [M_i \ddot{q}_i - g_i] = 0.
$$

(B.1)

Here, the $\delta u_i$, $i = l, l+1, \ldots, n-1$, are kinematically admissible for the joint between bodies $B_l$ and $B_{n-1}$, and the $\delta u_i$, $i = n+1, n+2, \ldots, m$, must be kinematically admissible for tree structure constraints in chain $\gamma_2$. The virtual displacements $\delta q_n$ are arbitrarily. Therefore, the coefficient of $\delta q_n$ must be zero and the following expression for the acceleration $\ddot{q}_n$ of the relative joint coordinates is obtained

$$
\ddot{q}_n = -(B_i^T [M_i B_i]^{-1} B_i^T (M_n (A_n \ddot{q}_n - B_n \ddot{q}_n + c_n) - g_n + \phi_n^T \lambda)).
$$

(B.2)

Substitution of equation (B.2) back into (B.1) yields

$$
\sum_{i=1}^{n-1} \delta u_i^T [M_i \ddot{q}_i - g_i] + \delta u_{n+1}^T [(M_{n+1} + \bar{N}_{n+1}) \ddot{q}_{n+1} - (g_{n+1} + \bar{g}_{n+1}) + \bar{\phi}_{n+1}^T \lambda] \\
+ \delta u_{m+2}^T [M_{m+2} \ddot{q}_{m+2} - g_{m+2} + \phi_{m+2}^T \lambda] + \sum_{i=m+3}^{m} \delta u_i^T [M_i \ddot{q}_i - g_i] = 0.
$$

(B.3)
for kinematically admissible virtual displacements $\delta u_i$, where $\hat{\mathbf{M}}_{n,i}$, $\hat{\mathbf{g}}_{n,i}$ and $\hat{\phi}^T_{n,i}$ are defined by

$$
\hat{\mathbf{M}}_{n,i} := A_i^T (I - M_i B_i (B_i^T M_i B_i)^{-1} B_i^T) M_i A_i,
\hat{\mathbf{g}}_{n,i} := -A_i^T (I - M_i B_i (B_i^T M_i B_i)^{-1} B_i^T) [M_i c_i - g_i],
\hat{\phi}^T_{n,i} := A_i^T (I - M_i B_i (B_i^T M_i B_i)^{-1} B_i^T) \phi^T_{n,i}.
$$

The accelerations of body $B_i$ have now been eliminated. This procedure can be repeated along chain $\gamma_1$ down to an arbitrary body $B_{k+1}$. By induction it can be shown that, we obtain

$$
\sum_{i=1}^{k-1} \delta u_i^T [M_i \ddot{v}_i - g_i] + \delta u_{k+1}^T [M_{k+1} \ddot{v}_{k+1} - (g_{k+1} + \hat{\mathbf{g}}_{k+1}) + \hat{\phi}_{k+1}^T \lambda] + \delta u_{n+2}^T [M_{n+2} \ddot{v}_{n+2} - (g_{n+2} + \hat{\mathbf{g}}_{n+2}) + \hat{\phi}_{n+2}^T \lambda] + \sum_{i=n+2}^{m} \delta u_i^T [M_i \ddot{v}_i - g_i] = 0,
$$

(B.4)

for kinematically admissible $\delta u_i$, $i = l, l+1, \ldots, k+1$, and $\delta u_i$, $i = n+1, n+2, \ldots, m$, for the tree structure constraints in chains $\gamma_1$ and $\gamma_2$, respectively. The equations (2.25) and (2.27), relating the virtual displacements and the accelerations of body $B_i$ and $B_j$ with $i$ and $j$ replaced by $k$ and $k+1$, respectively, can be substituted into equation (B.4), giving

$$
\sum_{i=1}^{k-1} \delta u_i^T [M_i \ddot{v}_i - g_i] + \delta u_{k+1}^T [M_{k+1} \ddot{v}_{k+1} - (g_{k+1} + \hat{\mathbf{g}}_{k+1}) + \hat{\phi}_{k+1}^T \lambda] + \delta u_{n+2}^T [M_{n+2} \ddot{v}_{n+2} - (g_{n+2} + \hat{\mathbf{g}}_{n+2}) + \hat{\phi}_{n+2}^T \lambda] + \sum_{i=n+2}^{m} \delta u_i^T [M_i \ddot{v}_i - g_i] = 0.
$$

(B.5)

for $\delta u_i$, $i = l, l+1, \ldots, k$, that are kinematically admissible for joints between bodies $B_i$ to $B_k$. As before, $\delta u_{k+1}$ is arbitrary and for that reason the coefficient of $\delta u_{k+1}$ must be zero. This results in

$$
\ddot{q}_{k+1} = -(B_{k+1}^T (M_{k+1} + \hat{\mathbf{M}}_{k+1}) B_{k+1})^{-1} B_{k+1}^T [M_{k+1} \ddot{v}_{k+1} + \hat{\mathbf{g}}_{k+1} + \hat{\phi}_{k+1}^T \lambda] - (g_{k+1} + \hat{\mathbf{g}}_{k+1}) + \hat{\phi}_{k+1}^T \lambda.
$$

(B.6)

Substitution of equation (B.6) into (B.5) gives

$$
\sum_{i=1}^{k-1} \delta u_i^T [M_i \ddot{v}_i - g_i] + \delta u_{k+1}^T [M_{k+1} \ddot{v}_{k+1} - (g_{k+1} + \hat{\mathbf{g}}_{k+1}) + \hat{\phi}_{k+1}^T \lambda] + \sum_{i=n+2}^{m} \delta u_i^T [M_i \ddot{v}_i - g_i] = 0,
$$

for kinematically admissible $\delta u_i$, $i = l, l+1, \ldots, k+1$, and $\delta u_i$, $i = n+1, n+2, \ldots, m$, for the tree structure constraints in chains $\gamma_1$ and $\gamma_2$, respectively. The equations (2.25) and (2.27), relating the virtual displacements and the accelerations of body $B_i$ and $B_j$ with $i$ and $j$ replaced by $k$ and $k+1$, respectively, can be substituted into equation (B.4), giving

$$
\sum_{i=1}^{k-1} \delta u_i^T [M_i \ddot{v}_i - g_i] + \delta u_{k+1}^T [M_{k+1} \ddot{v}_{k+1} + \hat{\mathbf{g}}_{k+1} + \hat{\phi}_{k+1}^T \lambda] + \sum_{i=n+2}^{m} \delta u_i^T [M_i \ddot{v}_i - g_i] = 0.
$$

(B.7)
where

\[ \bar{M}_k := A_{k+1}^T (I - (M_{k+1} + \bar{M}_{k+1})B_{k+1}(B_{k+1}^T (M_{k+1} + g_{k+1})B_{k+1})^{-1}B_{k+1}^T (M_{k+1} + \bar{M}_{k+1})A_{k+1}). \]

\[ g_k := -A_{k+1}^T (I - (M_{k+1} + \bar{M}_{k+1})B_{k+1}(B_{k+1}^T (M_{k+1} + \bar{M}_{k+1})B_{k+1})^{-1}B_{k+1}^T (M_{k+1} + \bar{M}_{k+1})A_{k+1}) \]

\[ \tilde{\Phi}_k := A_{k+1}^T (I - (M_{k+1} + \bar{M}_{k+1})B_{k+1}(B_{k+1}^T (M_{k+1} + \bar{M}_{k+1})B_{k+1})^{-1}B_{k+1}^T). \]

This procedure can be performed for each chain in the spanning tree from each cut joint body to the first junction body encountered and will result in the following equation for the junction body.
APPENDIX B. REDUCTION OF EQUATIONS OF MOTION FOR A CLOSED LOOP SYSTEM
RELATION BETWEEN THE
FUNDAMENTAL SOLUTION
AND THE GREEN FUNCTION
OF AN ODE AND A SLIGHTLY
PERTURBED ONE

Consider the IVPs
\[ y' = Ay + f, \quad \text{with} \quad y(t_0) = \beta, \quad (C.1) \]
and the perturbed IVP
\[ z' = \bar{A}z + \bar{f}, \quad \text{with} \quad z(t_0) = \beta, \quad (C.2) \]
where
\[ \bar{A} = A + \delta A, \quad \text{and} \quad \bar{f} = f + \delta f. \]
Then (C.2) yields
\[ z' = Az + f + \delta Ax + \delta f, \quad \text{with} \quad z(t_0) = \beta. \]
Let \( \Phi, \tilde{\Phi} \) denote the fundamental solutions of (C.1), (C.2), respectively. Then, the solution of (C.2) is given by
\[
x(t) = \Phi(t)\beta + \int_{t_0}^{t} \Phi(t)\tilde{\Phi}^{-1}(s)\bar{f}(s)ds \\
= \Phi(t)\beta + \int_{t_0}^{t} \Phi(t)\tilde{\Phi}^{-1}(s)(\delta A\beta + f(s) + \delta f(s))ds \\
= \Phi(t)\beta + \int_{t_0}^{t} \Phi(t)\tilde{\Phi}^{-1}(s)(\delta A\beta + \bar{f}(s))ds \\
\]
As a consequence,
\[ \Phi(t) = \Phi(t) + \int_{t_0}^{t} \Phi(t)\tilde{\Phi}^{-1}(s)\delta A\beta \tilde{\Phi}(s)ds. \quad (C.3) \]
Let $G(t, s)$ and $\tilde{G}(t, s)$ denote the Green functions of (C.1) and (C.2), respectively. Then

$$\tilde{G}(t, s) = G(t, s) + \Phi(t)\tilde{G}(0, t) - \int_t^s G(t, s) \delta A(s) \tilde{G}(t, s) \, ds. \tag{C.4}$$

With the definition

$$|\delta A| \leq \varepsilon_A,$$

the aforementioned relations make it possible to compare the conditioning constants of the two neighbouring problems (C.1) and (C.2). If $\kappa_2 \varepsilon_A < 1$, then

$$\bar{k}_1 \leq \frac{\kappa_1}{1 - \kappa_1 \varepsilon_A - \kappa_2 \varepsilon_A}, \quad \bar{k}_2 \leq \frac{\kappa_1}{1 - \kappa_1 \varepsilon_A - \kappa_2 \varepsilon_A}. \tag{C.5}$$

This means that a well-conditioned IVP remains reasonably conditioned if $\mathbf{A}$ and the initial conditions are perturbed only slightly. From relations (C.3) and (C.4) we can conclude that $\Phi$ and $\tilde{\Phi}$ may differ a lot if $||\delta A||$ is large. The same holds for $G$ and $\tilde{G}$. 
REFERENCES


[78] K. TAUBERT. Differenzverfahren für Schwingungen mit trockener und zäher Rei-

[79] A. F. VERESHCHAGIN. Computer simulation of the dynamics of complicated

[80] R. A. WEHAGE AND E. J. HAUG. Generalized coordinate partitioning for dimen-
sion reduction in analysis of constrained mechanical systems. J. of Mechanical
Design 104 (1982).

[81] P. M. E. J. WIUCKMANS. Multibody systems. RANA Report of the Department of
Mathematics and Computing Science 92-10, Eindhoven University of Technology,

[82] P. M. E. J. WIUCKMANS. Differential algebraic equations. RANA Report of the
Department of Mathematics and Computing Science 93-02, Eindhoven University

[83] P. M. E. J. WIUCKMANS AND R. M. M. MATTHEIJ. Discontinuities in mechan-
ical systems. In Proceedings of the Sixth Colloquium on Differential Equations

INDEX

$O(n)$—formulations, 25
absolute angular velocity, 13
absolute coordinates, 5
absolute position, 13
absolute variables, 5
acceleration constraints, 17
ADAMS, 18
algebraic variables, 37
augmentation method, 5, 11
backward difference formulae, 41
base body, 20
Baumgarte parameters, 86
Baumgarte stabilization, 86
BDF, 41
body connection array, 20
canonical projector, 50
center of mass, 13
chain, 21
close to higher index DAE, 60
closed chain, 6
closed loop, 19, 26
closed loop system, 6, 11
composite generalized coordinates, 15
composite generalized load, 15
composite generalized mass, 15
conditioning, 10, 12, 49, 51
conditioning constants, 51
configuration, 14
consistent initial conditions, 32
constrained equations of motion, 17
constrained variational equations of motion, 15
constraint, 4
  holonomic constraint, 15
  nonholonomic constraint, 15
constraints
  acceleration constraints, 17
  position constraints, 17
  velocity constraints, 17
coordinate projection methods, 89
cut joint, 6, 7, 11
cut joints, 21
d’Alembert’s principle of virtual work, 15
DADS, 18
DAE, 6, 8, 31
derivative projection methods, 89
descriptor form, 5, 11, 17
differential algebraic equation, 6, 8, 31
differential index, 11, 35
differential variables, 37
direct approach, 43
drift, 85, 87
dynamic analysis, 2
dynamic variable, 52
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges, 20</td>
<td></td>
</tr>
<tr>
<td>effective index, 64</td>
<td></td>
</tr>
<tr>
<td>elimination method, 5, 11</td>
<td></td>
</tr>
<tr>
<td>equations of motion, 16</td>
<td></td>
</tr>
<tr>
<td>father, 20</td>
<td></td>
</tr>
<tr>
<td>fundamental solution, 50</td>
<td></td>
</tr>
<tr>
<td>generalized applied loads, 15</td>
<td></td>
</tr>
<tr>
<td>generalized constraint loads, 15</td>
<td></td>
</tr>
<tr>
<td>generalized coordinate partitioning, 88</td>
<td></td>
</tr>
<tr>
<td>generalized coordinates, 7, 14</td>
<td></td>
</tr>
<tr>
<td>generalized loads</td>
<td></td>
</tr>
<tr>
<td>generalized applied loads, 15</td>
<td></td>
</tr>
<tr>
<td>generalized constraint loads, 15</td>
<td></td>
</tr>
<tr>
<td>generalized singular value decomposition,65</td>
<td></td>
</tr>
<tr>
<td>generating polynomials, 41</td>
<td></td>
</tr>
<tr>
<td>GSVD, 65</td>
<td></td>
</tr>
<tr>
<td>Hessenberg form, 45</td>
<td></td>
</tr>
<tr>
<td>holonomic constraint, 15</td>
<td></td>
</tr>
<tr>
<td>implicit DAE, 36</td>
<td></td>
</tr>
<tr>
<td>implicit Runge-Kutta, 46</td>
<td></td>
</tr>
<tr>
<td>index, 9, 11</td>
<td></td>
</tr>
<tr>
<td>differential index, 11</td>
<td></td>
</tr>
<tr>
<td>perturbation index, 11</td>
<td></td>
</tr>
<tr>
<td>index of nilpotency, 33</td>
<td></td>
</tr>
<tr>
<td>indirect approach, 43</td>
<td></td>
</tr>
<tr>
<td>inertia matrix, 13</td>
<td></td>
</tr>
<tr>
<td>IRK, 46</td>
<td></td>
</tr>
<tr>
<td>joint, 14</td>
<td></td>
</tr>
<tr>
<td>joint coordinates, 6</td>
<td></td>
</tr>
<tr>
<td>kinematically admissible displacement, 15</td>
<td></td>
</tr>
<tr>
<td>Lagrange equations of type two, 4</td>
<td></td>
</tr>
<tr>
<td>Lagrange multiplier, 4, 6, 11, 16</td>
<td></td>
</tr>
<tr>
<td>Lagrange's equations of type one, 17</td>
<td></td>
</tr>
<tr>
<td>local index, 35</td>
<td></td>
</tr>
<tr>
<td>machine precision, 92</td>
<td></td>
</tr>
<tr>
<td>manifold of discontinuity, 108</td>
<td></td>
</tr>
<tr>
<td>mass, 13</td>
<td></td>
</tr>
<tr>
<td>matrix pencil, 32</td>
<td></td>
</tr>
<tr>
<td>method of Lagrange multipliers, 11, 16</td>
<td></td>
</tr>
<tr>
<td>multibody formalism, 4</td>
<td></td>
</tr>
<tr>
<td>multibody system, 1</td>
<td></td>
</tr>
<tr>
<td>multistep method, 11, 40</td>
<td></td>
</tr>
<tr>
<td>nodes, 20</td>
<td></td>
</tr>
<tr>
<td>nonholonomic constraint, 15</td>
<td></td>
</tr>
<tr>
<td>ODAE, 90</td>
<td></td>
</tr>
<tr>
<td>ODE, 6</td>
<td></td>
</tr>
<tr>
<td>open loop, 19</td>
<td></td>
</tr>
<tr>
<td>open loop system, 6, 11</td>
<td></td>
</tr>
<tr>
<td>ordinary differential equations, 6</td>
<td></td>
</tr>
<tr>
<td>overdetermined DAE, 90</td>
<td></td>
</tr>
<tr>
<td>perturbation index, 11, 36</td>
<td></td>
</tr>
<tr>
<td>PIRK, 47</td>
<td></td>
</tr>
<tr>
<td>position constraints, 17</td>
<td></td>
</tr>
<tr>
<td>primary joints, 21</td>
<td></td>
</tr>
<tr>
<td>principle of virtual work, 15</td>
<td></td>
</tr>
<tr>
<td>Projected Implicit Runge-Kutta methods, 47</td>
<td></td>
</tr>
<tr>
<td>projection methods, 89</td>
<td></td>
</tr>
<tr>
<td>projector, 50</td>
<td></td>
</tr>
<tr>
<td>recursive formalism, 7</td>
<td></td>
</tr>
<tr>
<td>recursive formulation, 19</td>
<td></td>
</tr>
<tr>
<td>recursive Jacobian of constraints, 27</td>
<td></td>
</tr>
<tr>
<td>recursive load, 25, 27</td>
<td></td>
</tr>
<tr>
<td>recursive mass, 25, 27</td>
<td></td>
</tr>
<tr>
<td>regularization method, 86</td>
<td></td>
</tr>
<tr>
<td>relative joint coordinates, 18</td>
<td></td>
</tr>
<tr>
<td>relative variables, 5, 6</td>
<td></td>
</tr>
<tr>
<td>resultant moment, 14</td>
<td></td>
</tr>
<tr>
<td>Runge-Kutta method, 11</td>
<td></td>
</tr>
<tr>
<td>secondary joints, 21</td>
<td></td>
</tr>
<tr>
<td>semi-explicit DAE, 36, 51</td>
<td></td>
</tr>
<tr>
<td>shift operator, 40</td>
<td></td>
</tr>
<tr>
<td>singularity detecting code, 109</td>
<td></td>
</tr>
<tr>
<td>spanning tree, 7, 20</td>
<td></td>
</tr>
<tr>
<td>ssf, 39</td>
<td></td>
</tr>
<tr>
<td>ssf-solution, 91</td>
<td></td>
</tr>
<tr>
<td>stability, 50</td>
<td></td>
</tr>
</tbody>
</table>
INDEX

stabilizing Lagrange multipliers, 89
state equation, 50
state space equation, 4, 5, 11
state space form, 39
state space representation, 18
state variable, 5, 11, 50
stretched variable, 62
switching function, 107
system graph, 20

total force, 14
total virtual work, 15
tree configuration, 6
tree structure, 20
tree structure joints, 21
underlying ODE, 35
uniform index one DAE, 43
UODE, 35, 39, 61

variational equations of motion, 14
velocity constraints, 17
virtual displacement, 15

Weierstrass-Kronecker canonical form, 32
SAMENVATTING

Het doel van dit proefschrift is de ontwikkeling van betrouwbare en efficiënte numerieke methoden voor het simuleren van de dynamica van multibody systemen met gesloten ketens. De dynamica van deze systemen wordt beschreven door differentiaalvergelijkingen met bronfuncties welke vaak aan discontinuïteiten onderhevig zijn. Dit betekent dat numerieke methoden voor het oplossen van de bewegingsvergelijkingen van multibody systemen zodanig ontworpen moeten zijn, dat ze op een efficiënte en robuuste manier de dynamica van dergelijke mechanische systemen kunnen simuleren.

Multibody systemen zijn mechanische systemen bestaande uit starre dan wel flexibele lichamen, welke op dusdanige wijze verbonden zijn dat er grote relativie bewegingen tussen de lichamen onderling kunnen optreden. In dit proefschrift hebben we ons beperkt tot starre lichamen. De bewegingsvergelijkingen voor deze systemen kunnen op verschillende manieren worden opgesteld.

Een eerste manier is de augmentatie methode, waarbij de posities en de bewegingen van de lichamen met behulp van globale coördinaten worden beschreven. In deze methode worden de algebraïsche constraintvergelijkingen aan de dynamische vergelijkingen (dit zijn differentiaalvergelijkingen) toegevoegd, hetgeen resulteert in een stelsel van differentiaal-algebraïsche vergelijkingen (DAEs).

Een tweede methode is de eliminatie methode, waarbij men een relatieve beschrijvingswijze gebruikt. Door het gebruik van relatieve (onafhankelijke) variabelen kunnen de verbindingen en de verbindingskrachten worden geëlimineerd. In het geval van "open loop" systemen (waarbij alle lichamen op een unieke manier met elkaar zijn verbonden) betekent dit dat het dynamisch gedrag van multibody systemen door een stelsel van gewone differentiaalvergelijkingen (ODEs) wordt beschreven. Bij "closed loop" systemen zullen sommige lichamen op meer dan één manier met elkaar verbonden zijn. Voor dit soort systemen zal ook de eliminatie methode leiden tot een systeem van DAEs omdat er, vergeleken met "open loop" systemen, extra verbindingen aanwezig zijn.

In het algemeen zullen de bewegingsvergelijkingen van multibody systemen dus be-
Schreven worden door een stelsel van DAEs, welke overigens ook in andere toepassingen veelvuldig voorkomen.

Vaak hebben DAEs een speciale structuur waarbij de differentiaalvergelijkingen en de algebraïsche vergelijkingen afzonderlijk opgeschreven kunnen worden. In dit geval spreekt men van een semi-expliciete DAE. Multibody systemen zijn semi-expliciet. Bij nader onderzoek blijkt de wijze waarop de zogenaamde algebraïsche variabelen van de algebraïsche vergelijking afhangen van de toestandsvariabelen essentieel. Indien deze algebraïsche variabelen bijvoorbeeld via een inverse direct in de toestandsvariabelen uitgedrukt kunnen worden, spreekt men van een index één DAE. Op soortgelijke wijze kunnen ook DAEs van index twee en hoger geïdentificeerd worden.

In dit proefschrift hebben we allereerst de conditioning (dat wil zeggen de gevoeligheid van de oplossing voor storingen in de vergelijkingen) van semi-expliciete DAEs van index één en index twee onderzocht. Daar bij index één DAEs de algebraïsche variabelen via matrix inverse uitgedrukt kunnen worden in de toestandsvariabelen, ligt hier een duidelijk probleem indien deze matrix, die verder met D aangeduid zal worden, vrijwel singulier zal worden. Voor index twee DAEs blijkt een iets ingewikkelder inverse een analoge, maar essentieel ingewikkelder probleem op te roepen. Conditioneringsresultaten uit de literatuur suggereren dat enkele stabiliteitsconstanten afhankelijk van de norm van de inverse van deze matrix D, zodat deze constanten groot zullen worden als D vrijwel singulier wordt. Dit betekent dat er als gevolg van kleine storingen in de beginwaarden en de bronfuncties grote storingen in de oplossing van dit soort problemen kunnen ontstaan. In de hoofdstukken 4 en 5 hebben we echter aangetoond dat deze DAEs zich effectief kunnen gedragen als DAEs van een hogere index. Als we veronderstellen dat deze hogere index systemen zelf goed geconditioneerd zijn, dan kunnen we aannemen dat de oorspronkelijke DAEs (van lagere index) ook goed geconditioneerd zijn in de zin van eerstgenoemde zijn, de conditioneringsconstanten zijn dan niet groot. Het blijkt echter dat de conditioneringsconstanten wel groot kunnen worden voor dit soort DAEs (met vrijwel singuliere matrix D) wanneer geen hoger index systeem is aan te geven dat hier dicht bij ligt. Een interessant fenomeen is bovendien dat kleine storingen in de coëfficiënten van DAEs van index twee en hoger kunnen leiden tot grote veranderingen in de oplossingsgeschriften van deze DAEs, dit in tegenstelling tot ODEs en DAEs van index één.

De index van een DAE blijkt een maat te zijn voor de problemen die optreden bij het numeriek oplossen van DAEs. In het algemeen convergeren numerische methoden slechts voor DAEs van index één en kan er orde-reductie optreden in de zogenaamde algebraïsche variabelen. Numerieke methoden gebaseerd op achterwaartse differenties, zogenaamde BDF methoden, blijken gelukkig te convergeren voor grote klassen van systemen. Hiertoe horen ook de bewegingsvergelijkingen van multibody systemen, welke overigens DAEs van index drie zijn. Het blijkt dat discretisatie van dit soort systemen met behulp van BDF methoden echter resulteren in slecht geconditioneerde iteratier XVI. Dit impliceert dat afstandsfouten en fouten ontstaan door linearisatie zullen worden doorgegeven, waarbij ze vermenigvuldigd worden met negatieve machten van de stapgrootte $h$ en wel met $h^{-m+1}$ voor DAEs van index $m$. De conditioning van deze iter-
tienmatrices is uitgebreid bestudeerd in hoofdstuk 6. Er is gebleken dat voor DAEs van lagere index, die zich effectief gedragen als DAEs van een hogere index, de conditie-
ringkoosten van de uit de discretisaties resulterende iteratiematrices zich gedragen als bij systemen van hogere index. Dit heeft tot gevolg dat het effect van fouten op de
oplossing bij dit soort systemen groter kan zijn dan hetgeen verwacht zou mogen wor-
den op grond van hun index. De consequentie daarvan is dat BDF codes met variabele
stapgrootte en variabele orde niet meer goed werken, zelfs voor DAEs van index één. We
hebben een methode uitgewerkt waarbij we dit soort problemen voorkomen, hetgeen be-
tekent dat DAEs van hogere index rechtstreeks (dat wil zeggen zonder de index te redu-
ceren) opgelost kunnen worden. Het gebruik hiervan gecombineerd met een projectie-
methode heeft geleid tot een efficiënte numerieke methode voor het oplossen van DAEs
van hogere index. Deze methode is met succes toegepast op enkele multibody systemen.

Tenslotte, zoals opgemerkt, kunnen er bij de modellering van mechanische systemen
vaak discontinuïteiten optreden. Hierbij kan men bijvoorbeeld denken aan systemen die
een botsing ondergaan en aan systemen waar Coulombse wrijving aanwezig is, waarbij
de overgang van het systeem van de glijfase in de plakfase en vice versa aanleiding geeft
tot discrete overgangen. Deze discontinuïteiten manifesteren zich als eindige spron-
gen in de bronfuncties zelf of als sprongen in afgeleiden van deze functies. Het opferden
van deze discontinuïteiten geeft aanleiding tot ernstige moeilijkheden tijdens het nume-
riek oplossen van de bewegingsvergelijkingen voor multibody systemen. Er wordt een
methode uitgewerkt hoe met deze discontinuïteiten om te gaan. Deze wordt geïllustreerd
aan de hand van enkele mechanische voorbeelden.
CURRICULUM VITAE
