Index and Characteristic Analysis of Partial Differential Equations

by

Wade Steven Martinson

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2000

Author . . . .
Department of Chemical Engineering
19 November, 1999

Certified by . . . .
Paul I. Barton
Associate Professor of Chemical Engineering
Thesis Supervisor

Accepted by . . . .
Robert Cohen
St. Laurent Professor of Chemical Engineering
Chairman, Department Committee on Graduate Students

Index and Characteristic Analysis of Partial Differential Equations

by

Wade Steven Martinson

Submitted to the Department of Chemical Engineering
on 19 November, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

Technologies for dynamic simulation of chemical process flowsheets, as implemented in equation-based dynamic simulators, allow solution of fairly sophisticated process models, that include detailed descriptions of physical phenomena along with operating policies and discrete events. Simulation of flowsheet models with this level of detail forms the basis for a wide variety of activities, such as process optimization, startup and shutdown studies, process design, batch policy synthesis, safety interlock validation, and operator training. Technologies that make these activities possible for plant-scale models include sparse linear algebra routines, robust backward difference formula time integration methods, guaranteed state event location algorithms, generation of analytical Jacobian information via automatic differentiation, efficient algorithms for consistent initialization that may also be used to analyze the index of the model equations, automatic index reduction algorithms, and path-constrained dynamic optimization methods.

An equation-based dynamic process simulator takes as input the model equations that describe process behavior, along with a description of the operating policy. The input language allows for model decomposition, inheritance, and reuse, which facilitates construction of plant-scale dynamic models. Technologies like the ones mentioned above allow the simulator to then analyze the model for inconsistencies and perform calculations based on dynamic simulation, with a minimum of intervention from the engineer. This reduces both the time and numerical expertise required to perform simulation-based activities. Results, in some cases made possible or economically feasible only by the modeling support provided by a simulator, have been impressive.

However, these capabilities apply to flowsheet models that consist only of differential-algebraic, or *lumped*, unit models. Sometimes behavior in a particular unit cannot be adequately described by a lumped formulation, when variation with other independent variables like distance along a PFTR, film coordinate, or polymer chain length are important. In this case, behavior is most naturally modeled with partial differential, or *distributed*, unit models.

Partial differential equations in network flow simulations bring an additional set of mathematical and numerical issues. For a distributed model to be mathematically well-posed, proper initial and boundary conditions must be specified. Boundary condition requirements for nonlinear unit models may change during the course of a dynamic simulation, even in the absence of discrete events. Some distributed models, due to improper formulation or simple transcription errors, may be ill-posed because they do not have a mathematical property called *continuous dependence on data*. Finally, the model equations must be discretized in the proper manner.

This thesis contributes two new analyses of distributed unit models. The first relies on the definition of a differentiation index for partial differential equations developed in this thesis. It is by design a very natural generalization of the differentiation index of differential-algebraic equations. As such, and in contrast with other indices defined very recently for partial differential equations, it allows algorithms that are already used by process simulators to automatically analyze lumped unit models to be applied in a very straightforward manner to distributed unit models as well.

This index analysis provides insight into the equations that constrain consistent Cauchy data, which is the multidimensional analogue of initial data for differential-algebraic equations. It also provides an indication of the expected index of the differential-algebraic equations that result from method of lines semidiscretizations.

The second contribution of this thesis is an analysis of the mathematical well-posedness of distributed unit models provided by the engineer. This analysis relies on a generalization of the classical characteristic analysis of hyperbolic systems to more general nonhyperbolic systems, also developed in this thesis. It depends on the generalized eigenvalues and eigenvectors of a matrix pair, or alternatively on the (stably computable) transformation of a matrix pair to its generalized upper-triangular form. Because those quantities may be readily computed, this analysis may also be performed by a process simulator.

The analysis provides detailed information about the boundary conditions required to guarantee existence and uniqueness of the solution. It provides information about the smoothness required of forcing functions and data in order to guarantee a smooth solution to a linear system, or that is necessary for existence of a smooth solution to a nonlinear system. It also identifies systems that are ill-posed because they do not depend continuously on their data.

The ultimate goal for distributed models in dynamic process simulators is the level of support currently available for lumped models, where an engineer can provide an arbitrary model and expect the simulator to return a solution that is accurate to within a specified tolerance. It is unreasonable to expect a simulator to return a meaningful result if a distributed model is not mathematically well-posed. By identifying such models and offering information on how to make them well-posed, the analyses developed in this thesis allow a simulator to reduce the time and expertise required to set up and perform dynamic simulations involving distributed models.

Thesis Supervisor: Paul I. Barton
Title: Associate Professor of Chemical Engineering

To Brenda

Acknowledgments

I cannot say enough about what a pleasure working with Paul has been. Paul is a truly bright and gifted researcher. His enthusiasm is infectious. He is quick to spot the value in and new applications for the work of his students. He has no patience for second-class research, but is able to walk the fine line that separates constructive criticism of results from destructive criticism of the student. The research described in this thesis is without question the best I could do, and no small amount of the credit belongs to Paul.

Thanks also must go to my family - my father Steve, mother Karen, brother Ryan, and sisters Cheryl and Kari. My parents over the years have sacrificed in countless ways for the benefit of myself and my brother and sisters. I hope they are as proud of me as I am of each and every one of them.

I sometimes think of my grandfathers, Ellworth Theodore "E. T." Carlstedt and Marlow Martinson, both deceased, and wish that I could deliver a simple message to each of them. To E. T., who was unable to pursue his Ph.D. after completing his Master's degree, I would say, "You finally have a doctor in the family." And to Marlow, my message would simply be, "The oldest son of your oldest son hopes you are proud."

MIT can be a difficult place, particularly in the competitive atmosphere of the first year in the doctoral program. Fortunately, Jeff White was among the incoming graduate students. We shared many similar adjustments and experiences during that first year process, and his encouragement made a huge difference. Jeff has been and continues to be the best friend anyone could ever want.

Thank you also to the good friends I have met, or in the case of Eric Gallagher really got to know well, here in Boston - Eric and Yoli Gallagher, Andy Dahley, Ha Bui, Mark and Kelly Smith, Mark Fokema, and Julie Sherman. I am fortunate to have had good friends for coworkers as well. In particular I would like to thank John Tolsma, Arvind Mallik, and David Collins, for helping to make my stay in Boston such a great experience. Also, I would like to thank Phil Gallagher and Ryan and

Kevin Skogg, friends who, though separated by distance, I still consider among my best.

I came to MIT because I truly enjoy learning, which has come from my parents but also from the wonderful teachers I have had over the years. A complete list of the excellent teachers I've had would fill a page or more, but in particular I would like to mention Jim Grinsel, Mark Olbrantz, Pauline Babler, and Karen Stoddard of the Wausau School District, Art Guetter, Wojciech Komornicki, Olaf Runquist, and Jerry Artz at Hamline University, and Prodromos Daoutidis at the University of Minnesota.

I would also like to thank the staff of the MIT Department of Chemical Engineering, who have helped me in a million little ways during my stay. In particular, Craig Abernethy and Elaine Aufiero have helped me time and time again take care of problems or find the answers to my questions about the Institute.

Thanks to the person or people who I have inevitably (and regrettably) omitted as I try to properly write my acknowledgments. A pessimist like me always assumes the worst, and there is nothing worse than failing to properly recognize another person for their help.

Finally, I dedicate this thesis to the one person who has shared it all with me - my wife Brenda. She is the love of my life and my best friend. We are an incredible team, and this work is truly the product of a team effort.

Contents

List of Figures

Chapter 1

Dynamic Systems Modeling

1.1 Introduction

A chemical process flowsheet may be viewed as a network of unit operations, or units. Classical unit operations include distillation, continuously stirred tank reactors (CSTRs), plug-flow tubular reactors (PFTRs), and heat exchangers. The abstraction of a wide variety of industrial processes to a smaller set of basic unit operations marked a significant change in the chemical engineering profession.

A dynamic model of a process flowsheet is typically built from this unit operation paradigm. The engineer describes the behavior of each unit in the flowsheet with a particular model. Coupling the inputs and outputs of the individual unit models then produces the overall flowsheet model. For example, setting the exit stream from a reactor model equal to one of the input streams to a heat exchanger model reflects a plant structure in which the reactor effluent is fed to a heat exchanger.

The operating policy forms another layer of the flowsheet model. Specific conditions may trigger discrete events, such as the rupture of a bursting disk when the pressure in a vessel exceeds a critical level, or the injection of catalyst into a batch reactor at a certain time. A chemical process flowsheet is therefore most naturally modeled as such a hybrid discrete-continuous system [5].

Dynamic simulation of chemical plants forms the basis for a wide variety of activities [5, 53] such as process optimization [26], startup and shutdown studies [75], process design [53], batch policy synthesis [2], safety interlock validation [65], and operator training [83]. However, dynamic simulation of a chemical process flowsheet also presents significant challenges, which arise due to the size and complexity of the behavior that the dynamic model must capture.

Construction of the model requires significant effort. The size of the overall model contributes to this, because development of good models for each unit in the flowsheet demands good quantitative understanding of each unit's behavior. Additional work is required to couple the unit models together and specify degrees of freedom and forcing functions for the differential equations in such a way that the resulting dynamic simulation problem is mathematically well-posed. These conditions may need to be altered or revised during the course of a simulation. Finally, a great deal of numerical expertise is required to set up and debug a dynamic simulation involving large, sparse systems of differential-algebraic equations.

Modern chemical process simulators attempt to address these issues in several ways. First, they reduce the time required to create the flowsheet model through model decomposition, inheritance, and reuse. This means that an engineer may leverage models of behavior that are shared by several units. For example, a basic CSTR model might include overall material and species balance equations. A jacketed CSTR might use the same material and species balance equations, but also include a description of a steam jacket. A modern simulator allows the jacketed CSTR model to inherit the properties of the simple CSTR, so that the only additional modeling work required to create the jacketed CSTR model is description of the steam jacket.

Second, process simulators reduce the level of numerical expertise required to set up and debug a dynamic simulation through the use of advanced robust solution algorithms, which can be applied to general flowsheet models that consist of differential-algebraic equations. Examples of such solution algorithms include implicit, or backward difference formula (BDF), adaptive time step integration routines [28, 39, 68], automatic generation of Jacobian information via automatic differentiation [84], and guaranteed state event location via interval arithmetic [64]. The goal is to solve any mathematically well-posed flowsheet model automatically, without intervention by the engineer.

Third, they provide some automated analysis of the model equations. Examples include degree of freedom analysis to determine whether the number of equations matches the number of unknowns, structural analysis to detect ill-posed algebraic systems [23] and decompose their solution [82], structural index analysis to help specify consistent initial conditions [63, 71], reinitialization after discrete events [45, 57], and automatic reformulation of high index problems [25, 26, 27, 56]. For example, consider a flowsheet model that consists of 100,000 equations but is missing a single initial condition. One can imagine how identification, by the simulator, of a subset of five variables from which one initial condition will form a well-posed problem can greatly reduce the amount of time required to get the simulation working.

Process simulators thus have a fairly sophisticated and effective set of capabilities that reduce the time, effort, and numerical expertise required to perform simulation-based activities. Results, in some cases made possible or economically feasible only by the modeling support provided by these tools, have been impressive [16, 34, 51, 75].

However, these capabilities apply to flowsheet models that consist of differential-algebraic, or *lumped*, unit models. Sometimes behavior in a particular unit cannot be adequately described by a lumped formulation, when variation with other independent variables like distance along a PFTR, film coordinate, or polymer chain length are important. In such a case, behavior is most naturally modeled with partial differential, or *distributed*, unit models.

Partial differential equations in network flow simulations bring an additional set of mathematical and numerical issues. For a distributed model to be well-posed, proper initial and boundary conditions and forcing functions must be specified. Due to the connections with other units, boundary condition requirements for nonlinear unit models may change during the course of a dynamic simulation, even in the absence of discrete events. The model equations must be discretized in the proper manner, in order to generate a numerical solution.

The problem of proper discretization of a particular unit model has been the subject of an immense volume of research, which will be discussed in more detail in the next section. Comparatively little research has been directed toward development of automatable model analysis tools, however. Efforts of which this author is aware are also discussed in the next section.

The contribution of this thesis is two analyses of distributed unit models. The first is an index analysis, inspired by index analysis of lumped models, that provides insight into consistent initial and boundary conditions, as well as the index of semidiscretizations of distributed models. The second is a generalization of classical characteristic analysis of hyperbolic equations to nonhyperbolic systems, which provides insight into whether or not a given distributed model and its initial and boundary conditions form a well-posed problem. Both of these analyses may be performed automatically by a chemical process simulator.

1.2 Technology review

A significant body of research has been devoted to developing discretization methods that are tailored to particular models (see, for example, [19, 32, 47, 72, 73, 79, 87]). These methods are ideally suited to many repeated simulations of a single unit, where the mathematical properties of the model are very well understood. They tend to be inflexible, however, in that a scheme tailored for use with one set of equations may not be readily applicable to a different set of equations.

Support for simulations involving general distributed models falls into several categories. Most are again built for the applied mathematician or engineer interested in a single domain. These tools may be further divided into two types - library routines for discretization and integration, and high-level modeling languages. There are also the process simulators designed for systems engineers. Support for distributed unit models in these process simulators is still very limited.

1.2.1 Library routines for general PDEs

Many PDE packages for single-domain models consist of library routines. These typically consist of pieces of FORTRAN code and documentation for interfacing them with user supplied routines.

One of the earliest of these packages was PDECOL [52]. This was a collection of 19 FORTRAN subroutines, designed to solve a system of N equations over one spatial dimension of the form

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}(t, x, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_{xx}) \tag{1.1}$$

where

$$\begin{aligned} \mathbf{u} &= (u_1, u_2, \ldots, u_N) \\ \mathbf{u}_x &= (\frac{\partial u_1}{\partial x}, \frac{\partial u_2}{\partial x}, \ldots, \frac{\partial u_N}{\partial x}) \\ \mathbf{u}_{xx} &= (\frac{\partial^2 u_1}{\partial x^2}, \frac{\partial^2 u_2}{\partial x^2}, \ldots, \frac{\partial^2 u_N}{\partial x^2}) \end{aligned} \tag{1.2}$$

The user supplied FORTRAN routines that defined $\mathbf{f}$, boundary conditions, and initial conditions. PDECOL transformed (1.1) into a system of ODEs using collocation on finite elements, and then integrated these ODEs forward in time using an implicit backward differentiation routine for stiff systems. The user provided an array of element boundaries, and specified the polynomial order used for the elements. Initial and boundary conditions were supplied by the user. It was the user's responsibility to define a mathematically meaningful PDE problem.

EPDECOL [42] is a version of PDECOL that uses sparse linear algebra routines. These routines are faster than the solvers implemented in PDECOL. The authors report savings of 50 percent or more in total execution time using the sparse routines. It does not include any changes to the form of PDEs that can be solved using the package.

Another library of routines for the solution of PDEs is FIDISOL [77]. This package is designed for nonlinear elliptic or parabolic equations of the form

$$P(t, x, y, z, u, u_t, u_x, u_y, u_z, u_{xx}, u_{yy}, u_{zz}) = 0 \tag{1.3}$$

on a rectangular domain. The user must supply boundary and initial conditions. The package then uses variable order finite difference approximations for all spatial derivative terms. The selection of rectangular domains and finite differences were required to vectorize the algorithms. Again, the user is responsible for supplying a properly posed mathematical formulation.

SPRINT [6] is a collection of routines for solution of mixed systems of time dependent algebraic, ordinary differential, and partial differential equations. The partial differential equations are provided by the user in terms of a master equation format given by

$$\begin{aligned}\sum_{p=1}^{N} C_{j,p}(x,t,\mathbf{u},\mathbf{u}_x,\mathbf{v})\frac{\partial u_p}{\partial t} + Q_j(x,t,\mathbf{u},\mathbf{u}_x,\mathbf{v},\mathbf{v}_t) = \\ x^{-m}\frac{\partial}{\partial x}(x^m R_j(x,t,\mathbf{u},\mathbf{u}_x,\mathbf{v})), \quad j=1,\ldots,N\end{aligned} \tag{1.4}$$

where m is an integer which denotes the space geometry type. SPRINT provides routines for lumped finite element or collocation on finite element spatial discretization. The resulting ODEs are then considered together with the rest of the differential-algebraic equations, and integrated in time. Four time integration routines are provided in the package. The user must select the spatial and temporal discretization schemes. It is again up to the user to provide a well posed problem.

PDE/Protran [76] is a finite element-based package designed to solve PDEs on a single domain over two independent variables. It admits up to nine partial differential equations, given in the form

$$\frac{\partial}{\partial x}A_i\left(u_i,\frac{\partial u_i}{\partial x},\frac{\partial u_i}{\partial y},\beta\right) + \frac{\partial}{\partial y}B_i\left(u_i,\frac{\partial u_i}{\partial x},\frac{\partial u_i}{\partial y},\beta\right) + F_i(u_i,\beta) = 0 \tag{1.5}$$

Here A, B, and F are possibly nonlinear functions, and u_i and β are an eigenvector and eigenvalue of the problem. The software calculates these eigenvalues and eigenvectors, and the values of the functions A and B. Applications for specific models, such as anisotropic waveguides [20], have been built using this package.

1.2.2 Dynamic process simulators

All of the packages described so far provide routines for discretization of the spatial domain and integration of the resulting differential-algebraic equations in time. They deal with a single domain. The engineer must provide only FORTRAN code for the model equations themselves. This reduces the expertise in numerical methods required to perform dynamic simulations. The mathematical form of the equations, initial conditions, and boundary conditions must be checked for consistency by the modeler.

The systems engineer models a network of coupled units. Each unit is described by a collection of equations. The unit connectivities are described by another set of equations. Provision of FORTRAN routines describing all units and all connections is very time consuming, so high level simulation languages have been developed that provide much greater flexibility and allow much more rapid model development, and are interfaced with solution algorithms.

One of the first high level equation-based simulation packages was COSY [13]. This simulator transforms the engineer's model of a combined discrete-continuous process into a set of FORTRAN calls to the GASP-V combined discrete-continuous simulation library [14]. COSY handles partial differential equations with a method of lines approach based on finite difference approximations to spatial derivatives, which are generated automatically.

Another early high level simulation packages was SpeedUp [66]. This language began as a FORTRAN program, and developed into a high level dynamic simulation package. Support for PDEs is not built into the package; all discretizations and boundary conditions must be formulated manually. There are no checks on boundary condition consistency. The system model may be decomposed into unit models.

Two other packages for dynamic simulation of systems models are DYMOLA [24] and its successor OMOLA [3]. These packages do not have built in support for PDEs, but do have a powerful connection and terminal concept. Information leaving or entering a unit is declared explicitly. This allows for consistency checks when the flowsheet is built from the submodels. However, the direction for that flow must also be declared explicitly, because it is part of the consistency checks. This can, as noted by the author, pose problems during a simulation, since this direction can vary with time.

Like COSY, gPROMS [5] is a high level simulation package that incorporates support for PDE semidiscretization into the modeling language itself. It will automatically generate discrete equations from PDEs as directed by the user [61, 62]. The choice of discretizations is expanded to include both simple finite differences and collocation on finite elements using regular grids, and the architecture is extensible to include other techniques. Boundary conditions must still be declared explicitly by the engineer, and no consistency checks are provided for these conditions.

1.2.3 Semidiscretization analysis tools

None of these high level packages assist with the tasks of selecting a suitable semidiscretization technique or picking the values like mesh spacing that are associated with a particular semidiscretization. Two approaches to this problem have been explored.

One approach involves creation of tools that facilitate rapid construction (and thus evaluation) of many different semidiscretizations. While COSY and gPROMS provide some capability in this area, both are limited to the schemes coded into the packages.

A more general tool for semidiscretization evaluation is TRIFIT [86]. This package defines a symbolic grid generation language for partial differential equations in one or two spatial dimensions. TRIFIT provides discretization operators, and the user defines spatial derivative terms and mesh refinements using these operators. The stability of the semidiscretization is then tested by performing a simulation using direct linear algebra and ODE integration routines that are built into the package (but are not described in the paper).

The GRIDOP package [49, 50] provides similar tools for generation of conservative finite difference schemes on logically rectangular domains in an arbitrary number of independent variables. The package takes as input a user-supplied definition of function spaces and associated scalar products, together with user-supplied definitions of grid operators as finite difference schemes. The user may then provide partial differential equations in terms of the defined grid operators or the adjoints of those operators, and the package returns the finite difference equations.

Somewhere between code libraries and semidiscretization analysis tools lies Diffpack [46]. Diffpack is a development framework for PDE-based computer simulation. This code library is fully object-oriented, with a well documented application programming interface. It contains a very wide variety of routines, including linear and nonlinear solvers, grid generation and refinement tools, finite element routines, and visualization support. The entire package is coded using C++ rather than FORTRAN, and is currently under very active development.

Another approach is to perform a formal analysis of a particular semidiscretization prior to using it in a simulation. PDEDIS [70] allows the user to symbolically specify a PDAE system of the form

$$\mathbf{n} + \mathbf{A}\frac{\partial^2 \mathbf{x}}{\partial z^2} + \mathbf{D}\frac{\partial \mathbf{x}}{\partial z} + \mathbf{E}\frac{\partial \mathbf{x}}{\partial t} + \mathbf{f}(\mathbf{x}) + \mathbf{g}(z,t) = 0 \tag{1.6}$$

with boundary and initial conditions. In general, the matrices $\mathbf{A}, \mathbf{D}, \mathbf{E}$ can be singular and may show functional dependencies of the form

$$\begin{aligned} \mathbf{A} &= \mathbf{A}(\frac{\partial \mathbf{x}}{\partial z}, \mathbf{x}, z, t) && (1.7) \\ \mathbf{D} &= \mathbf{D}(\mathbf{x}, z, t) && (1.8) \\ \mathbf{E} &= \mathbf{E}(\mathbf{x}, z, t) && (1.9) \end{aligned}$$

The nonlinear function $\mathbf{n}$ collects all terms not matching the functional form of any other term. PDEDIS then symbolically manipulates the equations into a standard form and characterizes it. This theoretically allows consistency and well-posedness of the model to be checked, although only basic consistency checks are implemented. These checks are not detailed in the paper.

The package also provides provides some analysis tools for spatial discretizations based on either finite differences or orthogonal collocation over the entire spatial domain using polynomial trial functions. It symbolically semidiscretizes the equations (1.6), and then linearizes the resulting DAE system about a reference state provided by the user, if it is not already linear. PDEDIS then specifies a set of grid points and produces a file for submission to MATLAB, where the eigenvalue spectrum is calculated. The user examines this spectrum to determine whether the behavior of the DAE system is acceptable. For example, if the user knows in advance that the system should decay over time, the eigenvalues of the discrete system should have real components less than zero. A scheme with a positive real eigenvalue could then be rejected in the preprocessing stage.

1.3 Motivating Examples

None of the libraries, simulators, or semidiscretization analysis tools discussed in the previous section are able to perform anything beyond a very rudimentary analysis of an engineer's distributed model itself. This section will introduce several cases that illustrate how the automatable model analysis tools developed in this thesis may be used to help a systems engineer perform dynamic simulations involving distributed unit models.

1.3.1 Larry's problem: pressure-swing adsorption

Larry works on greenhouse gas removal from a nitrogen gas stream by a two-column pressure swing adsorption process. The process flowsheet appears in figure 1-1. A continuous high pressure feed to the system is directed through one of the columns, where greenhouse gases are removed from the nitrogen stream by adsorption onto a zeolite packing. At the same time, a low pressure nitrogen stream is blown through the other column to remove the adsorbed species and carry them to another treatment unit. When the packing in the high pressure column approaches saturation, the high pressure feed is switched over to the second column, and the low pressure stream is switched to the first column. The process is repeated.

Larry's task is to improve the operating policy for the process. He plans to make use of dynamic simulation for as much preliminary work as possible, because the system cannot be taken offline without major expense. The lab has given him good values for the parameters in the Kikkinides and Yang model of pressure-swing adsorption processes [43], which describes column behavior under assumptions of isothermal operation, negligible axial dispersion and pressure drop, plug flow, instantaneous solid-gas phase equilibrium, and perfect gas behavior, all of which he judges to be

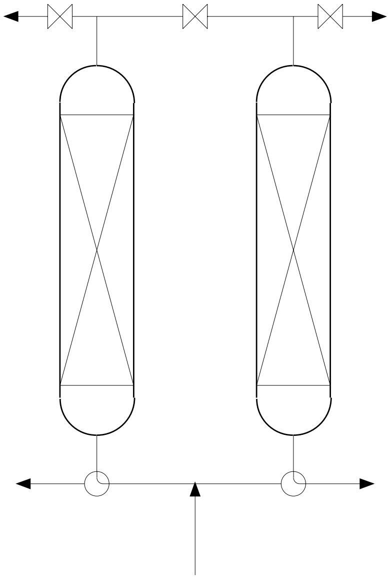

Figure 1-1: PSA flowsheet

reasonable for his process.

Under this model, the adsorbate concentration on the solid $q_{i=1\ldots3}$, mole fractions in the gas phase of adsorbate $y_{i=1\ldots3}$ and inert y_4, and flow velocity u are related by the following system of equations over time t and axial position in the absorber z. Pressure P, pressurization rate P_t, temperature T, bed void fraction ϵ, bed density ρ_B, gas constant R, saturation loadings $q^{sat}_{i=1\ldots3}$, and load relation correlation constants $n_{i=1\ldots3}$ and $B_{i=1\ldots3}$ are constant parameters. The values of these parameters have been experimentally verified for Larry's process.

$$\begin{aligned}
\frac{\rho_B RT}{P}\sum_{i=1}^{3} q_{i_t} + \frac{\epsilon}{P}P_t + u_z &= 0 \\
\epsilon y_{i_t} + \frac{\rho_B RT}{P} q_{i_t} + \frac{\epsilon y_i}{P} P_t + (u y_i)_z &= 0, \qquad i = 1\ldots3 \\
\sum_{i=1}^{4} y_i &= 1 \\
q_i - \frac{q_i^{sat} B_i (y_i P)^{\frac{1}{n_i}}}{1 + \sum_{j=1}^{3} B_j (y_j P)^{\frac{1}{n_j}}} &= 0, \qquad i = 1\ldots3
\end{aligned} \tag{1.10}$$

The first equation is the total material balance. The second equation is the material balances for the adsorbed species. The third equation forces the mole fractions in

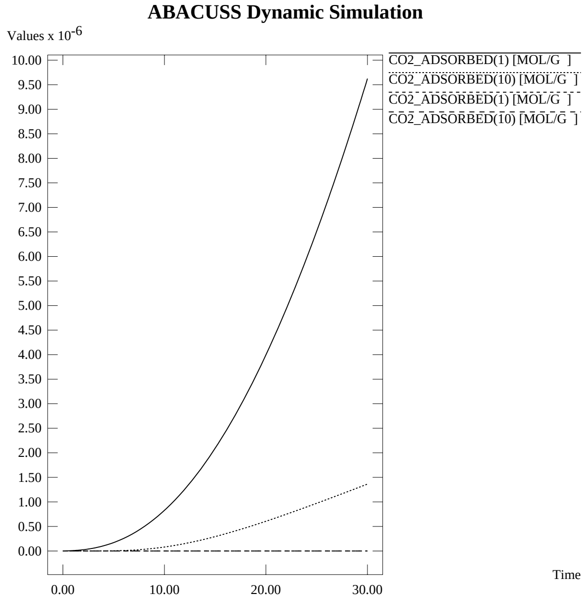

Figure 1-2: Simulation results for Larry's PSA problem

the gas phase to sum to unity. The fourth equation is the loading ratio correlations that give the equilibrium loading of each adsorbed component.

Larry needs to perform a dynamic simulation of the system from a cold start. Initial conditions for the six differential variables are

$$\begin{aligned} y_i(0,z) &= 1.0 \times 10^{-6}, \quad i = 1\ldots3 \\ q_i(0,z) &= 0, \quad i = 1\ldots3 \end{aligned} \tag{1.11}$$

while boundary conditions at startup are given by the feed compositions $y_{f,i=1\ldots3}$ and velocity $u_f = 0$.

$$\begin{aligned} y_i(t,0) &= y_{f,i}, \quad i = 1\ldots3 \\ u(t,0) &= u_f \end{aligned} \tag{1.12}$$

He uses a first order upwind finite difference scheme for spatial derivatives, and an implicit BDF integration method to advance the solution forward with t. The disappointing results appear in figure 1-2. The simulation fails after a simulated time of 30 seconds, when the reinitialization calculation required after the first valve position change does not converge.

What is wrong? The task facing Larry is to figure out what is wrong, and do it as quickly as possible.

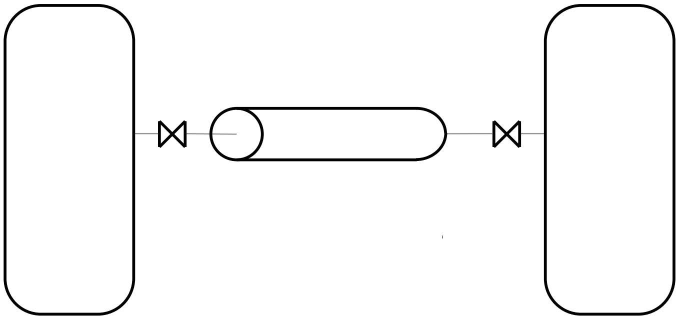

Figure 1-3: Vessel depressurization flowsheet

1.3.2 Moe's problem: compressible flow

Moe wants to simulate a vessel depressurization. The simplified flowsheet that he plans to use consists of two pressure vessels, two valves, and the process piping, and appears in figure 1-3. The gas is compressible, and if friction losses and gravity are ignored, radial variations are ignored, and the gas is assumed ideal, flow is described by the *Euler equations* [40, 72].

$$\begin{aligned} \rho_t + (\rho u)_x &= 0 \\ (\rho u)_t + \left(p + \frac{1}{2}\rho u^2\right)_x &= 0 \\ (\rho h)_t + (up - \rho u h)_x &= 0 \\ p &= (\gamma - 1)\rho i \\ h &= i + \frac{1}{2}u^2 \end{aligned} \tag{1.13}$$

Here ρ is the fluid density, u is the flow velocity, p is pressure, h is the specific total energy, and i is the specific internal energy. The first three model equations are conservation of mass, momentum, and energy respectively. The fourth is the ideal gas law, with a constant fluid heat capacity ratio of γ. The final equation relates total, internal, and kinetic energy.

The pipe segment under consideration is ten meters in length, so $0 \leq x \leq 10$, and

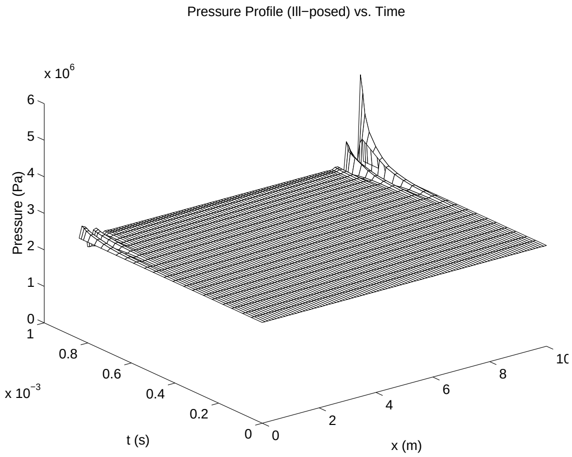

Figure 1-4: Pipe pressure profile

also let $t \geq 0$. The initial and boundary conditions are

$$\begin{aligned} \rho(0,x) &= 79.6 \ \text{kg/m}^3 \\ u(0,x) &= 0.0 \ \text{m/s} \\ p(0,x) &= 2.76 \ \text{MPa} \\ p(t,0) &= f_{valve1}(t) \\ p(t,10) &= f_{valve2}(t) \end{aligned} \tag{1.14}$$

The first scenario of interest to Moe is a case where the pressure in the pipe is initially slightly higher than the pressure in both vessels. The pressure in one vessel is significantly higher than the other.

Moe plans to solve the problem numerically using a first order upwind finite difference scheme [81]. He expects to initially see flow out of both ends of the pipe, followed by establishment of a steady pressure gradient and flow from the high pressure vessel to the low pressure vessel.

Simulation results, specifically the pressure profile along the pipe, appear in figure 1-4. Clearly, something is wrong. The calculated pressure profile blows up at the right endpoint. One would expect a rarefaction to enter the pipe from both ends, followed by establishment of a steady pressure gradient between the two ends. Instead, the calculated solution is blowing up after less than 0.3 simulated seconds.

Possible problems include improper boundary conditions, an improper discretization scheme, a time step or mesh spacing that is too large, and simple code bugs. Moe is faced with the task of uncovering the root of the problem and correcting it.

1.3.3 Curly's problem: electric power transmission

Curly works for a European power company. He needs to perform several simulations of 420kV power transmission lines. Current flow I and voltage with respect to ground u over a transmission line are described by the following simple system of two equations, which are known as the *telegrapher's equations.*

$$\begin{bmatrix} 0 & L \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ I \end{bmatrix}_t + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ I \end{bmatrix}_x + \begin{bmatrix} 0 & R \\ G & 0 \end{bmatrix} \begin{bmatrix} u \\ I \end{bmatrix} = 0 \tag{1.15}$$

Here L, C, R, and G are the inductance, capacitance, resistance, and conductance of the line per unit length.

The first scenario that Curly will simulate is a 1% increase in current demand occurring over 0.5 seconds, to be delivered over a 10 km line. For this particular line, $L = 0.0046$ Ω·s/km, $C = 6.5$ nF/km, $G = 33.3$ 1/Ω·km, and $R = 0.030$ Ω/km.

Measured values at the substation for AC power are 380 kV at 50 Hz, with a typical current demand of 3160 A. These values will be used for boundary conditions. The current demand will be given as a sinusoid increase from 3160 to 3192 over 0.5 seconds.

$$\begin{aligned} u(0,t) &= 190000 * \sin(50\pi t) \\ I(0,t) &= (1.0 + 0.005(1.0 + \sin(\pi(2t + 1.5))))3160 \end{aligned} \tag{1.16}$$

The domain is a ten kilometer line, and the simulation will cover the surge in demand, so $0 \leq x \leq 10$ and $0 \leq t \leq 0.5$.

Curly wants to build the complexity of the simulation slowly, so he begins with a simplified form [55] of the telegrapher's equations, that neglects the line inductance, resistance, and conductance.

$$\begin{bmatrix} 0 & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ I \end{bmatrix}_t + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ I \end{bmatrix}_x = 0 \tag{1.17}$$

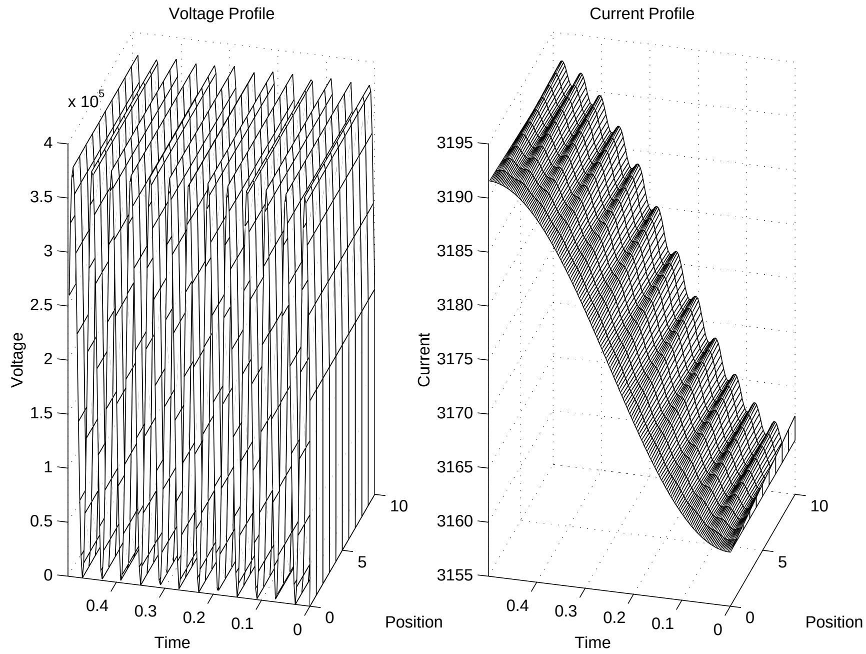

Figure 1-5: Simulation results for simplified electrical current model

While these assumptions behind this simplification are *not* valid for his system, experience with chemical process simulations has taught him to start with simplified models, and move to simulations based on more rigorous models once the simulation based on a simplified model is working.

He discretizes the partial derivative terms in x using centered finite differences, and initializes the line voltage to 190 kV. Simulation results for the simplified model appear in figure 1-5. The results look good, so he proceeds to the full model.

The partial derivative of current with respect to time, while absent from the simplified model, is present in the full model. Curly initializes the current in the line to its nominal demand of 3160 A. Results for the full current delivery model appear in figure 1-6. The simulation fails immediately.

1.3.4 Shemp's problem: combustion kinetics

Shemp works on combustion kinetics models for premixed diffusion flame propagation. His model of flame propagation uses four primary species, and assumes constant pressure, negligible radial gradients, and ideal gas behavior, and is based on a mole

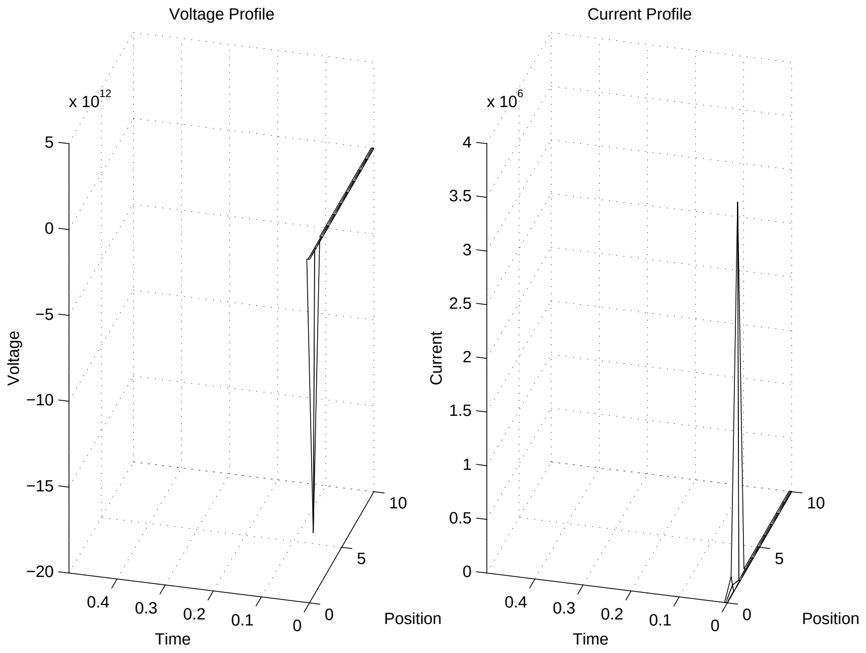

Figure 1-6: Simulation results for full electrical current model

balance formulation of the approach used by Miller et al [58].

$$
\begin{aligned}
c_{i_t} + (uc_i)_z + (c_i v_i)_z &= \omega_i \\
v_i &= -\frac{D_i}{x_i} x_{i_z} \\
x_i &= \frac{c_i}{\rho} \\
\omega_i &= f_i(T, c_j) \\
D_i &= h_i(y_i, T, x_j) \\
y_i \rho_m &= w_i c_i \\
\rho_m &= \rho w_{mean} \\
\rho &= \frac{P}{RT} \\
T &= g(z) \\
w_{mean} &= \frac{1}{\sum_{i=1}^{4} \frac{y_i}{w_i}} \\
u &= \frac{M}{\rho_m A}
\end{aligned}
\tag{1.18}
$$

The variables are the molar species concentrations c_i, diffusion velocities v_i, mole fractions x_i, net molar reaction rates ω_i, diffusion coefficients D_i, mass fractions y_i, mass density ρ_m, molar density ρ, temperature T, mean molecular weight w_{mean}, and flow velocity u. There are $6n + 5$ equations and variables. Parameters in the model are the flame cross-sectional area A, pressure P, gas constant R, mass flow rate M, and molecular weights w_i.

The first equation is a material balance on each species. The second gives the diffusion velocities. The fourth is the rate kinetics expression that gives the net molar production or consumption of each species per unit volume per unit time. The fifth gives a mixture-averaged diffusion coefficient based on binary diffusion coefficients. Other equations that relate the dependent variables should be self-explanatory.

Shemp plans to use dynamic simulation to fit parameters in his kinetic model. He plans to first provide the measured temperature profile and guesses for kinetic parameters, then calculate the steady-state solution to the model, and finally compare the calculated concentration profiles to measured profiles.

There are two basic approaches to obtaining a steady solution. One is to set the time derivatives immediately to zero, and employ a shooting method in z [38]. Other approaches have focused on integrating an implicit finite difference scheme for the time-dependent model to a steady state [67] or using finite differences to solve the steady boundary value problem directly [88].

Given recent advances in solution algorithms for DAEs discussed earlier in this chapter, including codes for efficient solution of large, sparse systems with BDF time integration, Shemp plans to revisit shooting methods for solution of the steady-state model. He starts at the cold end of the flame, using the following composition boundary conditions as initial conditions, and wants to integrate forward in z.

$$\begin{aligned}
y_1(0) &= 0.9979\\
y_2(0) &= 0.0001\\
y_3(0) &= 0.0010\\
y_4(0) &= 1.0 - y_1(0) - y_2(0) - y_3(0)\\
v_1(0) &= 0.0\\
v_2(0) &= 0.0\\
v_3(0) &= 0.0\\
v_4(0) &= 0.0
\end{aligned} \tag{1.19}$$

There are a total of 13 differential variables in z, so Shemp calculates the following additional values to be used as part of the boundary condition.

$$\begin{aligned}
x_1(0) &= 0.9543\\
x_2(0) &= 0.0075\\
x_3(0) &= 0.0010\\
x_4(0) &= 1.0 - x_1(0) - x_2(0) - x_3(0)\\
u(0) &= 15.412
\end{aligned} \tag{1.20}$$

The boundary conditions to be matched by shooting at the other end of the flame are

$$\mathbf{v}_z(L) = 0 \tag{1.21}$$

However, the simulation fails because the simulator cannot solve the consistent initialization problem. What is wrong?

1.4 Outline

The automatable model analysis tools developed in this thesis allow a process simulator to examine a distributed unit model automatically and help the engineer determine proper initial and boundary conditions, in order to form a mathematically well-posed problem. These tools also provide some insight into the expected smoothness of the solution and the index of a semidiscretization of the model. They can furthermore identify some models that will be ill-posed regardless of what initial and boundary conditions are provided, and thus cannot be solved as part of a dynamic simulation.

The next chapter provides a review of some of the mathematics on which these analyses are built. While most of the material is fairly basic, it is drawn from several very different areas, including linear algebra, abstract algebra, differential-algebraic equations, and partial differential equations. The presentation of this review material is designed to be approachable and easy to understand, rather than comprehensive, detailed, or completely rigorous. References listed at the beginning of the review chapter should be consulted for a more thorough treatment.

The following two chapters describe the two analyses developed during the project. The first is a differentiation index for partial differential equations. This index, unlike others that have been proposed for PDEs, is suitable for analysis of general distributed models by a process simulator. The second is a generalized characteristic analysis for nonhyperbolic systems. This analysis helps identify proper initial and boundary conditions for a distributed model, and identifies models that cannot be solved as part of a dynamic simulation.

The following chapter describes how these analyses may be performed by a process simulator. They will be applied to the problems facing Larry, Moe, Curly, and Shemp. The final chapter discusses the work so far, and examines what future efforts are needed.

Chapter 2

Math Review

This chapter contains a basic review of topics in linear algebra, abstract algebra, differential-algebraic equations, and partial differential equations. The linear algebra review is taken primarily from Strang [80] and Gantmacher [30]. The abstract algebra section is based on Fraleigh [29] and Aleksandrov et al. [1]. The basic material in the differential-algebraic equation section comes from Grossman [35], Campbell [10], and Brenan et al. [8]. The partial differential equation review is primarily taken from Courant and Hilbert [18], Jeffrey [40], and Lieberstein [48].

2.1 Linear Algebra

Linear algebra is the mathematics of linear systems of equations. Algebra begins with solution of a single equation for a single unknown, such as finding the value of x that satisfies

$$3x = 6 \tag{2.1}$$

The solution is found most simply by multiplying both sides by the inverse of the coefficient.

$$\begin{aligned} (3^{-1})3x &= (3^{-1})6 \\ x &= 2 \end{aligned} \tag{2.2}$$

In general, the solution to an equation of the form

$$ax = b \tag{2.3}$$

is given by

$$x = a^{-1}b \tag{2.4}$$

This solution exists and is unique[1] only if a^{-1} exists; that is, if $a \neq 0$.

Linear algebra considers systems of equations in several unknowns, such as

$$\begin{aligned} 3u + v + w &= 7 \\ 2u + 2v + 4w &= 12 \\ u + 3v + 2w &= 12 \end{aligned} \tag{2.5}$$

This system consists of only three equations in three unknowns, and already it takes considerably more space to write down than a single equation. Imagine writing a system of fifty, or a hundred, or 250,000 such equations!

2.1.1 Notation and operations

Clearly, linear algebra requires an efficient shorthand for writing systems of equations. The notation of linear algebra expresses a linear system (2.5) as the product of a **matrix** of coefficients **A** and a **vector** of unknowns **x**.

$$\mathbf{Ax} = \mathbf{b} \tag{2.6}$$

A system of any size may be written in this manner.

Matrices and vectors will be written in boldface type in this text. Matrices will be denoted by capital letters, while vectors will be lowercase letters. Other notations include underscores for matrices and vectors ($\underline{\underline{A}}$, $\underline{x}$), and overscored arrows for vectors $\vec{x}$.

[1] If $b = 0$, then any x satisfies the equation; if $b \neq 0$, no value for x can satisfy the equation.

The matrix **A** in the linear system (2.5) is an array of coefficients.

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 1 \\ 2 & 2 & 4 \\ 1 & 3 & 2 \end{bmatrix} \tag{2.7}$$

An individual **element** of a matrix is identified by its row, counted from the top, followed by its column, counted from the left. For example, $A_{23} = 4$. Notice also that A_{23} is a single number, or a **scalar**, so it is not written in boldface.

The vector **x** contains the unknowns.

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.8}$$

This vector may be thought of as a matrix that consists of three rows and one column. Such a vector, consisting of a single column, is sometimes called a **column vector**. A vector that consists instead of a single row is referred to as a **row vector**. In either case, an element of a vector is identified by its row or column, counted in the same way as a matrix. For example, $x_2 = v$.

Matrix-vector multiplication continues the convention of row-then-column. The first row of **A** is multiplied by the first (and only) column of **x**. Let $\mathbf{a}_1$ be a row vector equal to the first row of **A**.

$$\mathbf{a}_1 = \begin{bmatrix} 3 & 1 & 1 \end{bmatrix} \tag{2.9}$$

The **dot product** or **inner product** of two vectors is defined by adding the products of all corresponding elements of the two vectors.

$$\mathbf{a}_1 \cdot \mathbf{x} = 3u + v + w \tag{2.10}$$

Notice that the dot product of two vectors is a scalar.

Matrix-vector multiplication simply takes the dot product of the first row with the column vector, followed by the second row with the column vector, and so on.

The resulting scalars themselves form a column vector.

$$\mathbf{Ax} = \begin{bmatrix} - & \mathbf{a}_1 & - \\ - & \mathbf{a}_2 & - \\ - & \mathbf{a}_3 & - \end{bmatrix} \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \cdot \mathbf{x} \\ \mathbf{a}_2 \cdot \mathbf{x} \\ \mathbf{a}_3 \cdot \mathbf{x} \end{bmatrix} = \begin{bmatrix} 3u + v + w \\ 2u + 2v + 4w \\ u + 3v + 2w \end{bmatrix} \tag{2.11}$$

The righthand side $\mathbf{b}$ is also a column vector.

$$\mathbf{b} = \begin{bmatrix} 7 \\ 12 \\ 12 \end{bmatrix} \tag{2.12}$$

Setting this column vector equal to $\mathbf{Ax}$ is clearly just another way of writing the original system of equations (2.5):

$$\mathbf{Ax} = \begin{bmatrix} 3u + v + w \\ 2u + 2v + 4w \\ u + 3v + 2w \end{bmatrix} = \begin{bmatrix} 7 \\ 12 \\ 12 \end{bmatrix} = \mathbf{b} \tag{2.13}$$

Of course, the goal is to *solve* this linear system of equations. The similarity of the notation for the linear system (2.6) and a single equation (2.1) suggests that the solution should be given by multiplication of both sides of the system by the inverse of $\mathbf{A}$.

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{Ax} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \end{aligned} \tag{2.14}$$

Consideration of this inverse will be taken up later.

2.1.2 Solving a linear system

A system of equations is solved by transforming it into a set of individual equations, each in only one unknown, which may be solved like the first equation of this chapter (2.1). This approach of transforming a complicated problem into one or more simpler problems will be a continuing theme throughout this thesis.

As a demonstration of how a system of equations may be transformed into a set of individual equations in a single unknown, consider the following system.

$$\begin{aligned} 2x + 3y &= 8 \\ 4x + 11y &= 26 \end{aligned} \tag{2.15}$$

Solving systems like this one relies on two simple operations: adding equations together, and multiplying an equation by a scalar. Performing these two operations in the proper way transforms this linear system to a set of two equations, one involving only x and the other in only y.

The first step is multiplying both sides of the first equation by -2 to produce

$$-4x - 6y = -16 \tag{2.16}$$

and adding it to the second equation, which gives

$$5y = 10 \tag{2.17}$$

Already, multiplying an equation by a scalar and adding equations together has eliminated x from the second equation, producing a new equation in only one unknown. This may be solved by multiplying both sides by 5^{-1}, to produce

$$y = 2 \tag{2.18}$$

This equation (2.18) may be multiplied by -3 and added to the first equation in the original system (2.15), which gives

$$2x = 2 \tag{2.19}$$

Again, the solution is found by multiplying both sides of the equation by 2^{-1}.

$$x = 1 \tag{2.20}$$

Alternatively, once it is known that $y = 2$, one can simply substitute this back into the first equation, which gives

$$\begin{aligned} -4x - 6(2) &= -16 \\ -4x &= -4 \end{aligned} \tag{2.21}$$

This equation again gives the solution, which is $x = 1$.

The systematic process of repeatedly adding a multiple of one equation to a second equation in order to eliminate a variable from the second equation, ultimately producing a set of equations that each involve only one variable, is called **Gauss-Jordan elimination**. The process of systematically eliminating variables from equations until it is possible to solve the system through back substitution is called **Gauss elimination**.

2.1.3 Matrices and vectors

Addition of two matrices or two vectors is defined only if they are **conforming**, or of the same size. Let $\mathbf{x}$ and $\mathbf{y}$ be two conforming vectors. Each element in the sum $\mathbf{z} = \mathbf{x} + \mathbf{y}$ is the sum of the corresponding elements of $\mathbf{x}$ and $\mathbf{y}$; $z_i = x_i + y_i$. The same approach holds for the sum of two matrices.

Multiplication of two matrices is the same as multiplication of a matrix and a series of vectors. If the columns of a matrix $\mathbf{Z}$ are thought of as individual column vectors, the product $\mathbf{AZ}$ may be thought of as successive products of the matrix $\mathbf{A}$ and the columns of $\mathbf{Z}$:

$$\mathbf{AZ} = \mathbf{A} \begin{bmatrix} | & & | \\ \mathbf{z}_1 & \dots & \mathbf{z}_n \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbf{Az}_1 & \dots & \mathbf{Az}_n \\ | & & | \end{bmatrix} \tag{2.22}$$

Note that, in general, $\mathbf{AZ} \neq \mathbf{ZA}$.

The **identity** matrix $\mathbf{I}$ has unity on the diagonal and zero everywhere else:

$$\mathbf{I} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \tag{2.23}$$

$\mathbf{I}$ has the property that

$$\mathbf{IA} = \mathbf{AI} = \mathbf{A} \tag{2.24}$$

and also

$$\mathbf{I}\mathbf{x} = \mathbf{x} \tag{2.25}$$

The **transpose** of a matrix $\mathbf{A}$, denoted $\mathbf{A}^T$, is formed by transposing the row and column index of each entry, so $A^T_{ij} = A_{ji}$. The transpose of the coefficient matrix $\mathbf{A}$ (2.7) of section 2.1.1 is

$$\mathbf{A}^T = \begin{bmatrix} 3 & 2 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 2 \end{bmatrix} \tag{2.26}$$

Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors of the same length. The **projection of $\mathbf{x}$ onto $\mathbf{y}$** is given by

$$\frac{\mathbf{x} \cdot \mathbf{y}}{\mathbf{y} \cdot \mathbf{y}} \mathbf{y} \tag{2.27}$$

If the projection of $\mathbf{x}$ onto $\mathbf{y}$ is zero, then $\mathbf{x}$ and $\mathbf{y}$ are said to be **orthogonal**.

A collection of vectors is said to be **linearly independent** if

$$a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + \cdots = \mathbf{0} \;\; \Rightarrow \;\; a_i = 0 \tag{2.28}$$

for all i. If the vectors are not linearly independent, then one or more of them are simply a linear combination of the others.

A set of vectors defines a **subspace**, which consists of all vectors that may be formed by linear combinations of the vectors in the set. A set of independent vectors is called a **basis** for the subspace.

2.1.4 The determinant

Consider the coefficient a from the simple equation (2.3) given earlier. The solution x (2.4) exists and is unique iff $a \neq 0$. Now, let the **determinant** of a, written[2] as $|a|$, be simply the value of a. If the determinant is nonzero, clearly a may be inverted to

[2] $|a|$ refers to the determinant of a scalar a only in this section (2.1.4); elsewhere it will denote the magnitude (absolute value) of a.

determine a unique solution; if the determinant is zero, one cannot invert a, and no x satisfies the original equation (unless $b = 0$, in which case all x satisfy the original equation).

Now, move up from a scalar a to a square matrix $\mathbf{A}$, and define $|\mathbf{A}|$, the determinant of $\mathbf{A}$, recursively as follows: Pick some row i of $\mathbf{A}$, and multiply each element A_{ij} in the chosen row by the product of $(-1)^{i+j}$ and the determinant of the matrix formed by all elements of $\mathbf{A}$ *not* in row i or column j. The determinant of this smaller matrix is called the **cofactor** of A_{ij}.

For a 2×2 matrix, choose the first row. The determinant is then calculated using the above technique as

$$|\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}|a_{22}| - a_{12}|a_{21}| \tag{2.29}$$

Because $|a_{ij}| = a_{ij}$, the value of the determinant of a 2×2 matrix clearly does not depend on which row is chosen. This is also true for the determinant of matrices of any larger size.

Moving up to a 3×3 matrix, choose the first row again. Then $|\mathbf{A}|$ is given by

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \tag{2.30}$$

For example, consider the coefficient matrix (2.7) from section 2.1.1.

$$\begin{aligned} \begin{vmatrix} 3 & 1 & 1 \\ 2 & 2 & 4 \\ 1 & 3 & 2 \end{vmatrix} &= 3 \begin{vmatrix} 2 & 4 \\ 3 & 2 \end{vmatrix} - 1 \begin{vmatrix} 2 & 4 \\ 1 & 2 \end{vmatrix} + 1 \begin{vmatrix} 2 & 2 \\ 1 & 3 \end{vmatrix} \\ &= 3(-8) - 1(0) + 1(4) \\ &= -20 \end{aligned} \tag{2.31}$$

If $|\mathbf{A}| \neq 0$, then $\mathbf{A}$ is called **invertible** or **regular**. This means that it has an inverse $\mathbf{A}^{-1}$, and the associated system of equations (2.13) has a unique solution (2.14). If, however, $|\mathbf{A}| = 0$ is zero, then $\mathbf{A}$ is called **singular**, and it does not have an inverse. The parallel to the case of a single equation (2.3) and its solution (2.4) should be clear. Actual calculation of $\mathbf{A}^{-1}$ must wait a little longer; it will be considered in the next section.

If $\mathbf{A}$ is singular, there exists at least one vector $\mathbf{y} \neq \mathbf{0}$ such that

$$\mathbf{Ay} = \mathbf{0} \tag{2.32}$$

What is the significance of a square, singular matrix? Well, suppose $\mathbf{x}$ satisfies the system of equations given by

$$\mathbf{Ax} = \mathbf{b} \tag{2.33}$$

Because of the existence of $\mathbf{y}$, we know that $\mathbf{x}$ is not a unique solution, because $\mathbf{x}+\mathbf{y}$ also satisfies the equations:

$$\mathbf{A}(\mathbf{x}+\mathbf{y}) = \mathbf{Ax} + \mathbf{Ay} = \mathbf{Ax} + \mathbf{0} = \mathbf{b} \tag{2.34}$$

Suppose that there are p linearly independent vectors $\mathbf{y}_i$ that satisfy (2.32). The subspace generated by these vectors is called the **nullspace of A**. If $\mathbf{A}$ is an $n \times n$ matrix, the **rank** of $\mathbf{A}$ is $n - p$.

If $\mathbf{A}$ is invertible, then the value of $\mathbf{x}$ that satisfies $\mathbf{Ax} = \mathbf{b}$ is unique. Therefore

$$\mathbf{Ax} = \mathbf{0} \;\; \Rightarrow \;\; \mathbf{x} = \mathbf{0} \tag{2.35}$$

No nonzero vector satisfies (2.32), and the square matrix $\mathbf{A}$ is said to have **full rank**.

The determinant of the product of two matrices equals the product of the determinants.

$$|\mathbf{AB}| = |\mathbf{A}|\,|\mathbf{B}| \tag{2.36}$$

However, the same does not hold true for addition; the determinant of the sum of two matrices is *not* equal to the sum of the determinants.

$$|\mathbf{A}+\mathbf{B}| \neq |\mathbf{A}| + |\mathbf{B}| \tag{2.37}$$

2.1.5 Solution of linear systems revisited

Solving a system of equations via Gauss or Gauss-Jordan elimination relies on the operations of multiplying an equation by a scalar, and adding equations together. These operations may themselves be represented by matrices! Furthermore, these matrices are related to the inverse of the coefficient matrix $\mathbf{A}$.

Consider, for example, the simple system that was previously solved.

$$\begin{bmatrix} 2 & 3 \\ 4 & 11 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 8 \\ 26 \end{bmatrix} \qquad (2.38)$$

$$\mathbf{Az} = \mathbf{b}$$

Now, examine what happens when both sides of this equation are multiplied on the left by a special matrix $\mathbf{R}_1$.

$$\mathbf{R}_1\mathbf{Az} = \mathbf{R}_1\mathbf{b}$$

$$\begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 11 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 26 \end{bmatrix} \qquad (2.39)$$

$$\begin{bmatrix} 2 & 3 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$$

The first row of $\mathbf{R}_1\mathbf{A}$ is formed by the product of the row vector $[1\;0]$ and $\mathbf{A}$, which does not change the first row. The second row of $\mathbf{R}_1\mathbf{A}$ is the product of the row vector $[-2\;1]$ and $\mathbf{A}$. In words, the second row of the new matrix $\mathbf{R}_1\mathbf{A}$ is -2 times the first row of $\mathbf{A}$ plus the second row of $\mathbf{A}$. The net effect of multiplying the system on the left by $\mathbf{R}_1$ is that the first row of $\mathbf{A}$ remains unchanged, and the second row becomes the difference of the original second row and twice the first row.

Similarly, the other steps in the solution of the system may also be represented as matrices.

$$\begin{gathered}
\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{5} \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{5} \end{bmatrix} \begin{bmatrix} 8 \\ 10 \end{bmatrix} \\
\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 2 \end{bmatrix} \\
\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \\
\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}
\end{gathered} \tag{2.40}$$

As before, row operations have solved the system. The only difference is that the row operations have been expressed here as matrices:

$$\mathbf{R}_4\mathbf{R}_3\mathbf{R}_2\mathbf{R}_1\mathbf{A}\mathbf{z} = \mathbf{I}\mathbf{z} = \mathbf{R}_4\mathbf{R}_3\mathbf{R}_2\mathbf{R}_1\mathbf{b} \tag{2.41}$$

Now, recall that the inverse of $\mathbf{A}$ is a matrix $\mathbf{A}^{-1}$ that, when multiplied on the right by $\mathbf{A}$, produces the identity:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \tag{2.42}$$

The matrices that represent Gauss-Jordan elimination are this inverse! Let $\mathbf{R} = \mathbf{R}_4\mathbf{R}_3\mathbf{R}_2\mathbf{R}_1$. Then, because

$$\mathbf{R}\mathbf{A} = \mathbf{I} \tag{2.43}$$

it is clear that $\mathbf{R} = \mathbf{A}^{-1}$.

2.1.6 Matrix norms

The absolute value of a scalar x is a measure of its size.

$$|x| = (x \cdot x)^{\frac{1}{2}} \tag{2.44}$$

The positive root is always chosen, so $|x| \geq 0$.

A **norm** is a measure of the “size” of a vector or matrix [33, 80]. Perhaps the most common is called the **2-norm** or **Euclidean norm**; for a vector, it is given by

$$||\mathbf{x}||_2 = (\mathbf{x} \cdot \mathbf{x})^{\frac{1}{2}} \tag{2.45}$$

This norm satisfies the **Cauchy-Schwarz inequality**

$$|\mathbf{x} \cdot \mathbf{y}| \leq ||\mathbf{x}|| \, ||\mathbf{y}|| \tag{2.46}$$

and the **triangle inequality**

$$||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \tag{2.47}$$

Also, for any scalar a,

$$||a\mathbf{x}|| \leq |a| \, ||\mathbf{x}|| \tag{2.48}$$

A more general norm is the **p-norm** which, for a vector $\mathbf{x}$ with n elements, is given by

$$||\mathbf{x}||_p = \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}} \tag{2.49}$$

When a norm is written as simply $||\mathbf{x}||$, the 2-norm is often assumed.

The 2-norm of a matrix is defined using the 2-norm of a vector.

$$||\mathbf{A}|| = \max_{\mathbf{x} \neq \mathbf{0}} \frac{||\mathbf{A}\mathbf{x}||}{||\mathbf{x}||} \tag{2.50}$$

Because $||\mathbf{A}||$ is always greater than or equal to $||\mathbf{A}\mathbf{x}||/||\mathbf{x}||$ under this definition, it must be true that

$$||\mathbf{A}|| \, ||\mathbf{x}|| \geq ||\mathbf{A}\mathbf{x}|| \tag{2.51}$$

The 2-norm satisfies the **submultiplicative property**

$$||\mathbf{A}\mathbf{B}|| \leq ||\mathbf{A}|| \, ||\mathbf{B}|| \tag{2.52}$$

and also

$$||\mathbf{A} + \mathbf{B}|| \leq ||\mathbf{A}|| + ||\mathbf{B}|| \tag{2.53}$$

and finally, for a square $n \times n$ matrix $\mathbf{A}$,

$$\max_{i,j}|a_{ij}| \leq ||\mathbf{A}|| \leq n \max_{i,j}|a_{ij}| \tag{2.54}$$

2.1.7 Eigenvalues and eigenvectors

Every square matrix $\mathbf{A}$ has at least one special vector $\mathbf{y}$ that has the following property:

$$\mathbf{A}\mathbf{y} = \lambda \mathbf{y} \tag{2.55}$$

Multiplying this special vector $\mathbf{y}$ by $\mathbf{A}$ has the effect of simply scaling every element of $\mathbf{y}$ by the same constant factor λ. Such a vector is called an **eigenvector** of $\mathbf{A}$, and the scaling factor λ is called an **eigenvalue**.

Finding these eigenvectors and eigenvalues is a bit more complicated than simply solving a system $\mathbf{A}\mathbf{x} = \mathbf{b}$, because the righthand side is also unknown. One approach might be to move the righthand side over to the left. Then the system is

$$\mathbf{A}\mathbf{y} - \lambda \mathbf{y} = [\mathbf{A} - \lambda \mathbf{I}]\, \mathbf{y} = \mathbf{0} \tag{2.56}$$

and the new righthand side is now known. Clearly, $\mathbf{y} = \mathbf{0}$ is a solution to this system, for any value of λ, but it is the *nonzero* eigenvectors and the associated eigenvalues that are of interest.

Recall that if a matrix $\mathbf{M}$ is singular, there will be at least one nonzero vector $\mathbf{y}$ that satisfies $\mathbf{M}\mathbf{y} = \mathbf{0}$. For each value of λ such that $(\mathbf{A} - \lambda \mathbf{I})$ is singular, there will thus be at least one associated nonzero eigenvector.

A matrix is singular iff its determinant is zero, so what are needed are values of λ that give

$$|\mathbf{A} - \lambda \mathbf{I}| = 0 \tag{2.57}$$

This determinant is a polynomial in λ, and the roots of this polynomial are the eigenvalues of $\mathbf{A}$. Once the eigenvalues are known, the associated eigenvectors are simply the nonzero solutions of the original system (2.56).

An $n \times n$ matrix produces a polynomial of order n in λ, which will have exactly n roots. These roots are the eigenvalues of $\mathbf{A}$. Some of these roots may be zero. If the roots are all **distinct** (have different values), then all eigenvectors are linearly independent. In the case of n linearly independent eigenvectors, the matrix is said to have a **complete set of eigenvectors**. If a particular root λ_i is repeated j times, that eigenvalue has **algebraic multiplicity** j. If there are fewer than j linearly independent, nonzero eigenvectors that satisfy (2.56) for that particular root λ_i, then the matrix is said to be **deficient**.

As an example, consider the following 2 × 2 matrix.

$$\mathbf{A} = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \tag{2.58}$$

The eigenvalues are the roots of the polynomial given by

$$\begin{aligned} |\mathbf{A} - \lambda\mathbf{I}| &= \begin{vmatrix} 1-\lambda & 3 \\ 2 & 2-\lambda \end{vmatrix} \\ &= (1-\lambda)(2-\lambda) - (2)(3) \\ &= \lambda^2 - 3\lambda - 4 \end{aligned} \tag{2.59}$$

Factoring this polynomial and setting it equal to zero gives the eigenvalues.

$$\lambda^2 - 3\lambda - 4 = (\lambda - 4)(\lambda + 1) = 0 \;\;\Rightarrow\;\; \lambda_1 = 4, \lambda_2 = -1 \tag{2.60}$$

The eigenvectors satisfy $(\mathbf{A} - \lambda_i\mathbf{I})\mathbf{y}_i = \mathbf{0}$. The first eigenvector

$$\mathbf{y}_1 = \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.61}$$

must satisfy

$$\begin{bmatrix} 1-\lambda_1 & 3 \\ 2 & 2-\lambda_1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -3 & 3 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \;\;\Rightarrow\;\; a = b \tag{2.62}$$

Any nonzero choice may be made that satisfies $a = b$, so let

$$\mathbf{y}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{2.63}$$

A similar calculation for λ_2 gives

$$\mathbf{y}_2 = \begin{bmatrix} 3 \\ -2 \end{bmatrix} \tag{2.64}$$

Because such an eigenvector **y** is given by a system where the matrix is multiplied by **y** on its righthand side, it is sometimes called a **right eigenvector**. A **left eigenvector** is a row vector that satisfies

$$\mathbf{zA} = \lambda \mathbf{z} \tag{2.65}$$

Each eigenvalue λ is associated with an equal number of left and right eigenvectors. Unless otherwise noted, the term eigenvector will always refer to right eigenvectors.

2.1.8 Diagonalization and the Jordan form

Suppose that an $n \times n$ matrix $\mathbf{A}$ has a complete set of eigenvectors, and let the columns of $\mathbf{S}$ be those eigenvectors. Then

$$\mathbf{S}^{-1}\mathbf{AS} = \mathbf{\Lambda} \tag{2.66}$$

where $\mathbf{\Lambda}$ is a matrix with the eigenvalues of $\mathbf{A}$ on its diagonal and zero everywhere else. The eigenvectors **diagonalize** the matrix $\mathbf{A}$, and $\mathbf{A}$ is called **diagonalizable**[3].

One obvious application is calculating repeated powers of $\mathbf{A}$.

$$\mathbf{A}^p = \left(\mathbf{S\Lambda S}^{-1}\right)^p = \left(\mathbf{S\Lambda S}^{-1}\right)\left(\mathbf{S\Lambda S}^{-1}\right)\ldots\left(\mathbf{S\Lambda S}^{-1}\right) = \mathbf{S}\left(\mathbf{\Lambda}\right)^p\mathbf{S}^{-1} \tag{2.67}$$

Calculating the p^{th} power of a diagonal matrix simply requires the p^{th} power of each diagonal element, so this calculation is much easier than repeatedly multiplying $\mathbf{A}$ by itself.

Obviously it is not possible to diagonalize a matrix that does not have a complete set of eigenvectors. However, it is always possible to find an invertible matrix $\mathbf{M}$ for which

$$\mathbf{M}^{-1}\mathbf{AM} = \mathbf{J} \tag{2.68}$$

Here $\mathbf{J}$ is a **Jordan matrix**, and is called the **Jordan canonical form**, or simply the **Jordan form**, of $\mathbf{A}$. It is a block diagonal matrix. Each block on the diagonal

[3] Also, the rows of $\mathbf{S}^{-1}$ contain the complete set of *left* eigenvectors for the matrix.

has the form

$$\begin{bmatrix} \lambda_i & & & \\ 1 & \lambda_i & & \\ & \ddots & \ddots & \\ & & 1 & \lambda_i \end{bmatrix} \tag{2.69}$$

and is called a **Jordan block**. If every block has dimension 1, then $\mathbf{J} = \boldsymbol{\Lambda}$. A **lower Jordan matrix** is a matrix in this Jordan form; it is also possible to define the Jordan form as having the ones above the diagonal, which is called an **upper Jordan matrix**.

The columns of **M** are the **generalized eigenvectors** of **A**. Every such eigenvector $\mathbf{l}_i$ satisfies either

$$\mathbf{A}\mathbf{l}_i = \lambda_i \mathbf{l}_i \tag{2.70}$$

or

$$\mathbf{A}\mathbf{l}_i = \lambda_i \mathbf{l}_i + \mathbf{l}_{i+1} \tag{2.71}$$

Every square matrix has a complete set of these generalized eigenvectors, which are sometimes called a **chain** of eigenvectors. The number of eigenvectors in the chain is the **geometric multiplicity** of the corresponding eigenvalue. For a repeated root, the sum of the geometric multiplicities of the eigenvalue equals the algebraic multiplicity of the eigenvalue. For example, an eigenvalue of algebraic multiplicity 5 may have geometric multiplicities of 2 and 3, or 1 and 4, or 1 and 1 and 3, or 5, and so on.

2.1.9 Nilpotent matrices

A matrix **N** is **nilpotent**, or has **nilpotency k**, if

$$\mathbf{N}^k = \mathbf{0}, \quad \mathbf{N}^{k-1} \neq \mathbf{0} \tag{2.72}$$

Any upper triangular or lower triangular matrix where every diagonal element is zero is nilpotent. Also, every nilpotent matrix is singular.

If a particular eigenvalue λ_i of a matrix is zero, the corresponding Jordan block (2.69) is nilpotent and has the form

$$\mathbf{N} = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \tag{2.73}$$

The nilpotency of such a block is equal to the dimension of the block. Note, however, that the nilpotency of a particular block is not necessarily the nilpotency of the original matrix. The original matrix is only nilpotent if *every* block in its Jordan form is nilpotent.

2.1.10 The Drazin inverse

If the blocks in the Jordan form of a matrix **A** are ordered so that all blocks with $\lambda_i \neq 0$ appear above and to the left of all blocks with $\lambda_i = 0$, it has the form

$$\mathbf{J} = \begin{bmatrix} \mathbf{C} & \\ & \mathbf{N} \end{bmatrix} \tag{2.74}$$

Here **C** is invertible, while **N** is singular and nilpotent. Note that if **A** is invertible, the nilpotent block **N** disappears, while if **A** is nilpotent, the invertible block **C** disappears.

Because every square matrix **A** is equivalent to one in Jordan form, one can write **A** in terms of its Jordan form and generalized eigenvector matrix **M**.

$$\mathbf{A} = \mathbf{M}\mathbf{J}\mathbf{M}^{-1} = \mathbf{M} \begin{bmatrix} \mathbf{C} & \\ & \mathbf{N} \end{bmatrix} \mathbf{M}^{-1} \tag{2.75}$$

The **Drazin inverse** of **A** is then defined [10] as follows.

$$\mathbf{A}^D = \mathbf{M} \begin{bmatrix} \mathbf{C}^{-1} & \\ & \end{bmatrix} \mathbf{M}^{-1} \tag{2.76}$$

It has several important properties, including the fact that it commutes with **A**: $\mathbf{A}\mathbf{A}^D = \mathbf{A}^D\mathbf{A}$. Also, if 0 is an eigenvalue of **A** of algebraic multiplicity k, then 0 is also an eigenvalue of $\mathbf{A}^D$ of algebraic multiplicity k. Similarly if $\lambda_i \neq 0$ is an eigenvalue of $\mathbf{A}$ of algebraic multiplicity k_i, then $1/\lambda_i$ is an eigenvalue of $\mathbf{A}^D$ of algebraic multiplicity k_i.

Finally, note that the actual value of $\mathbf{A}^D$ is given here as the product of three matrices: $\mathbf{M}$, $\mathbf{M}^{-1}$, and the block diagonal matrix involving $\mathbf{C}^{-1}$ (2.76). Different choices of $\mathbf{M}$ may be employed to put $\mathbf{A}$ in a block diagonal form (2.75) with a different invertible $\mathbf{C}$ in the upper lefthand block and a different nilpotent $\mathbf{N}$ in the lower righthand block. These different choices produce different Drazin inverses. Once $\mathbf{M}$ is selected, however, the corresponding Drazin inverse is unique.

2.1.11 Matrix pairs and pencils

The set of all linear combinations of two matrices (assumed to be conforming) is called a **pencil**. A pencil is typically written as

$$\mathbf{B} - \lambda\mathbf{A} \tag{2.77}$$

although λ is sometimes taken to be the ratio of two scalars τ and ρ, with $\tau/\rho = \lambda$:

$$\rho\mathbf{B} - \tau\mathbf{A} \tag{2.78}$$

The latter expression allows "infinite" values of λ.

If every member of a particular pencil of square matrices is singular, the pair of matrices is said to be **singular**. The pencil is also called singular. If there is at least one combination of $\mathbf{A}$ and $\mathbf{B}$ that is invertible, the pair of matrices and the pencil are said to be **regular**.

2.1.12 Generalized eigenvectors and the Weierstrass form

A regular pencil of $n \times n$ matrices will have up to n singular members. These singular members are given by the pairs of scalars (ρ_i, τ_i) that are solutions of

$$|\rho_i\mathbf{B} - \tau_i\mathbf{A}| = 0 \tag{2.79}$$

Let (ρ_j, τ_j) be a pair of such scalars. Then, because $\rho_j \mathbf{B} - \tau_j \mathbf{A}$ is singular, there is a nonzero vector $\mathbf{x}_j$ such that

$$(\rho_j \mathbf{B} - \tau_j \mathbf{A})\, \mathbf{x}_j = \mathbf{0} \tag{2.80}$$

This vector and scalar pair are in a sense analogous to an eigenvector and eigenvalue of a single matrix, because

$$\rho_j \mathbf{B} \mathbf{x}_j = \tau_j \mathbf{A} \mathbf{x}_j \tag{2.81}$$

and are (somewhat confusingly) also called a **generalized eigenvector** and **generalized eigenvalue** of the matrix pair. A matrix pair may or may not possess a complete set of these generalized eigenvectors.

For every regular pair of matrices, there exist conforming invertible matrices $\mathbf{P}$ and $\mathbf{Q}$ such that

$$\mathbf{PAQ} = \begin{bmatrix} \mathbf{I} & \\ & \mathbf{N} \end{bmatrix} \quad \mathbf{PBQ} = \begin{bmatrix} \mathbf{J} & \\ & \mathbf{I} \end{bmatrix} \tag{2.82}$$

Here $\mathbf{J}$ is a lower Jordan matrix, and $\mathbf{N}$ is a lower Jordan nilpotent matrix. This is the **Weierstrass canonical form** of the matrix pair.

From this form, we define **doubly generalized eigenvectors** for the matrix pair, which are analogous to the generalized eigenvectors of a single matrix (2.70 - 2.71).

Lemma 2.1.1 *For every real-valued regular pencil of dimension n, there exist n generalized left eigenvectors $\mathbf{l}_i$ and eigenvalue pairs (ρ_i, τ_i) such that either $\tau_i \mathbf{l}_i \mathbf{A} = \rho_i \mathbf{l}_i \mathbf{B}$, or $\mathbf{l}_{i-1}\mathbf{A} + \tau_i \mathbf{l}_i \mathbf{A} = \rho_i \mathbf{l}_i \mathbf{B}$ with $\rho_i \neq 0$, or $\tau_i \mathbf{l}_i \mathbf{A} = \rho_i \mathbf{l}_i \mathbf{B} + \mathbf{l}_{i-1}\mathbf{B}$ with $\tau_i \neq 0$.*

Proof. Let $\mathbf{P}$ and $\mathbf{Q}$ be the matrices that transform $(\mathbf{A}, \mathbf{B})$ to their Weierstrass canonical form. Let $\mathbf{l}_i = \mathbf{p}_i$, where $\mathbf{p}_i$ is the i^{th} row of $\mathbf{P}$, and let $\rho_i = (\mathbf{PAQ})_{ii}$ and $\tau_i = (\mathbf{PBQ})_{ii}$. Then, by inspection, for every equation that corresponds to the first equation in a Jordan block of either $\mathbf{J}$ or $\mathbf{N}$,

$$\mathbf{l}_i \mathbf{AQ} = \begin{bmatrix} \dots & 0 & \rho_i & 0 & \dots \end{bmatrix} \quad \text{and} \quad \mathbf{l}_i \mathbf{BQ} = \begin{bmatrix} \dots & 0 & \tau_i & 0 & \dots \end{bmatrix}$$

By inspection $\tau_i \mathbf{l}_i \mathbf{AQ} = \rho_i \mathbf{l}_i \mathbf{BQ}$, so $\tau_i \mathbf{l}_i \mathbf{A} = \rho_i \mathbf{l}_i \mathbf{B}$.

For any other equation in the first block row, clearly $\rho_i \neq 0$, and

$$\mathbf{l}_i\mathbf{AQ} = \begin{bmatrix} \dots & 0 & \rho_i & 0 & \dots \end{bmatrix} \text{ and } \mathbf{l}_i\mathbf{BQ} = \begin{bmatrix} \dots & 1 & \tau_i & 0 & \dots \end{bmatrix}$$

Note also that

$$\mathbf{l}_{i-1}\mathbf{AQ} = \begin{bmatrix} \dots & \rho_i & 0 & 0 & \dots \end{bmatrix}$$

By inspection $\mathbf{l}_{i-1}\mathbf{AQ} + \tau_i\mathbf{l}_i\mathbf{AQ} = \rho_i\mathbf{l}_i\mathbf{BQ}$, so $\mathbf{l}_{i-1}\mathbf{A} + \tau_i\mathbf{l}_i\mathbf{A} = \rho_i\mathbf{l}_i\mathbf{B}$.

For any other equation in the second block row, $\tau_i \neq 0$, and

$$\mathbf{l}_i\mathbf{AQ} = \begin{bmatrix} \dots & 1 & \rho_i & 0 & \dots \end{bmatrix} \text{ and } \mathbf{l}_i\mathbf{BQ} = \begin{bmatrix} \dots & 0 & \tau_i & 0 & \dots \end{bmatrix}$$

Note also that

$$\mathbf{l}_{i-1}\mathbf{BQ} = \begin{bmatrix} \dots & \tau_i & 0 & 0 & \dots \end{bmatrix}$$

By inspection $\tau_i\mathbf{l}_i\mathbf{AQ} = \rho_i\mathbf{l}_i\mathbf{BQ} + \mathbf{l}_{i-1}\mathbf{BQ}$, so $\tau_i\mathbf{l}_i\mathbf{A} = \rho_i\mathbf{l}_i\mathbf{B} + \mathbf{l}_{i-1}\mathbf{B}$. □

Corollary 2.1.2 *For every real-valued regular pencil of dimension n, there exist n generalized right eigenvectors $\mathbf{r}_i$ and eigenvalue pairs (ρ_i, τ_i) such that either $\tau_i\mathbf{A}\mathbf{r}_i = \rho_i\mathbf{B}\mathbf{r}_i$, or $\mathbf{A}\mathbf{r}_{i+1} + \tau_i\mathbf{A}\mathbf{r}_i = \rho_i\mathbf{B}\mathbf{r}_i$ with $\rho_i \neq 0$, or $\tau_i\mathbf{A}\mathbf{r}_i = \rho_i\mathbf{B}\mathbf{r}_i + \mathbf{B}\mathbf{r}_{i+1}$ with $\tau_i \neq 0$.*

So, the rows of $\mathbf{P}$ and the columns of $\mathbf{Q}$ act in a manner that is analogous to the left and right generalized eigenvectors of a single matrix, respectively.

2.2 Abstract Algebra

A scalar and a matrix are related but different. Addition and multiplication are defined differently for the two, but yet there are some similarities, such as solution of an equation requiring invertibility of the coefficient, that are surely more than mere coincidence. Abstract algebra is the study of the precise definitions of the way calculations are performed on different types of mathematical objects.

2.2.1 Sets and binary operations

A collection of objects is called a **set**. Some authors take the concept of a set as a primitive concept, upon which other ideas are built, and so do not attempt to make the definition more precise [29]. Others may attempt to define a set based on simple physical examples like three apples, two oranges, and so forth [1]. Here the concept of set will be treated as a primitive.

The members of a set are called its **elements**; the set with no elements is called the **empty set** and is denoted by $\emptyset$. A set will be denoted by an italicized capital letter, such as S. Elements of a set will be denoted by italicized lower case letters. The notation $a \in S$ means that "a is an element of S". Important sets [29] include $\mathbb{R}$, the set of all real numbers, $\mathbb{C}$, the set of all complex numbers, $\mathbb{Z}$ (also denoted $\mathbb{N}$), the set of all integers, $\mathbb{Z}^+$, the set of all positive integers, and $\mathbb{N}_0$, the set of all nonnegative integers. $\mathbb{R}^{n \times n}$ is the set of all real-valued $n \times n$ matrices.

A set may be given as a rule that identifies its members. Let S be the set of all even integers. It may be written as $S = \{n \mid n/2 \in \mathbb{Z}\}$, which is read as "$S$ is the set of all elements n such that $n/2$ is an element of the integers".

A **binary operation** $*$ on a set S is a rule that assigns each ordered pair of elements of S to some element of S. The definition [29] incorporates the requirement that a binary operation on two elements of the set on which it is defined produces another element of that same set. This is called the **closure condition**; an operation is by definition not a binary operation on S if it does not meet this condition.

For example, let S be the set of all integers. Then addition is a binary operation on S, because the sum of any two integers is again an integer. Division is not a binary operation on S, because the ratio of two integers is not necessarily again an integer.

A binary operation $*$ on S is said to be **commutative** if and only if $a * b = b * a$ for all $a, b \in S$. The operation is called **associative** iff $(a * b) * c = a * (b * c)$ for all $a, b, c \in S$.

2.2.2 Groups

Upon the basic ideas of a set and of a binary operation defined on a set are built successively more complex concepts. The first of these is a **group**. A group $\langle G, * \rangle$ is a set G together with a binary operation $*$ on G for which

1. The binary operation $*$ is associative.

2. There is an element $e \in G$ (called the **identity element**, or simply the **identity**) such that $e * x = x * e = x$ for all $x \in G$.

3. For every $a \in G$, there is an element $a' \in G$ (called the **inverse of a with respect to** $*$) such that $a' * a = a * a' = e$.

A group is called **abelian** iff its binary operation is commutative.

2.2.3 Rings

The second concept or structure is called a **ring**. A ring $\langle R, +, \times \rangle$ is a set R together with two binary operations $+$ and $\times$ defined on R for which the following is true.

1. $\langle R, + \rangle$ is an abelian group (the identity element of this group is called **zero**, and the operation is called **addition**).

2. The operation $\times$ is associative.

3. For all $a, b, c \in R$ the **left distributive law**, $a \times (b + c) = a \times b + a \times c$, and the **right distributive law**, $(a + b) \times c = a \times c + b \times c$, hold.

If $\langle R, +, \times \rangle$ is a ring and $M_n(R)$ is the set of all $n \times n$ matrices on which the binary operations $+$ and $\times$ from $\langle R, +, \times \rangle$ are used to define corresponding operations on $M_n(R)$ (in the same manner as in section 2.1), it can be shown that $\langle M_n(R), +, \times \rangle$ is also a ring.

Note that a ring does not have a multiplicative inverse. Specifically, $\langle M_n(R), +, \times \rangle$ has an additive inverse but no multiplicative inverse. This means that it is possible to calculate determinants and perform Gauss elimination in $\langle M_n(R), +, \times \rangle$ using additive inverses to produce zeros, but it is not possible to reduce a diagonal matrix to a matrix with a multiplicative identity along the diagonal - there is no multiplicative identity!

2.2.4 Fields

The third structure is a **field**. A field $\langle F, +, \times \rangle$ is a set F together with two binary operations $+$ and $\times$ defined on F for which the following is true.

1. $\langle F, +, \times \rangle$ is a ring.
2. The operation $\times$ is commutative ($\langle F, +, \times \rangle$ is then called a **commutative ring**; also $\times$ is called **multiplication**).
3. There is a **multiplicative identity** $1 \in F$ such that $1 \times x = x \times 1 = x$ for all $x \in F$ (this identity element is called **unity**).
4. There is a **multiplicative inverse** in F for every *nonzero* element of F.
5. The operation $\times$ is commutative.

The real numbers, together with standard addition and multiplication, form a field, and unity refers to the number 1. Complex numbers also form a field with standard addition and multiplication.

Defining corresponding binary operations on the set $M_n(F)$ of $n \times n$ matrices whose elements belong to a field does not produce a field, but it does allow for (noncommutative) inverses over multiplication. Matrices of elements of a field, along with the corresponding matrix binary operations defined from the binary operations of that field, allow for matrix inverses, albeit ones that do not commute [29]. For example, the matrix $\mathbf{R}$ in section 2.1.5 is the *left* inverse of $\mathbf{A}$, because $\mathbf{RA} = \mathbf{I}$. The two matrices do not commute over multiplication, however, so $\mathbf{AR} \neq \mathbf{I}$.

2.2.5 Functions

Let D and C be any two sets. A **function** is a rule that matches elements of one set to elements of another. If a function f takes each element x of D and matches it to exactly one element y in C, then D is called the **domain** of f, and C is called the **codomain**. The function itself is written $f(x) = y$ or $f : D \to C$. The set R of all elements of C to which f matches one or more elements of D is called the **range** of f.

If, for every $x_1, x_2 \in D$ such that $x_1 \neq x_2$, $f(x_1) \neq f(x_2)$, then f is said to be **one-to-one**, or an **injection**. If for every $y \in C$ there exists an $x \in D$ such that $f(x) = y$, then f is said to be **onto**, or a **surjection**, and also the codomain is the same as the range ($C = R$).

If $\langle D, \times\rangle$ and $\langle C, \cdot\rangle$ are groups, a function $f : D \to C$ is called a **homomorphism** if

$$f(a \times b) = f(a) \cdot f(b) \tag{2.83}$$

for every $a, b \in D$. The function is an **isomorphism** if it is one-to-one and onto, and $\langle D, \times\rangle$ and $\langle C, \cdot\rangle$ are said to be **isomorphic**. Two groups that are isomorphic are completely equivalent; they are different notations for exactly the same algebraic structure.

The **inverse** of a function f is some other function f^{-1} such that, if $f(a) = b$, then $f^{-1}(b) = a$. This means that the composition of a function and its inverse return the original argument of the function; in other words, $f(f^{-1}(x)) = f^{-1}(f(x)) = x$. If a function is one-to-one and onto from D to R, that function has an inverse from R to D, and is called a **bijection**.

The parallel with matrices should be clear. If $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{A}$ has an inverse, then $\mathbf{A}^{-1}\mathbf{b} = \mathbf{x}$, and also $\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{x}$. Matrices are thus often though of as functions.

An **odd function** is one for which $f(-x) = -f(x)$. An **even function** is one for which $f(-x) = f(x)$. A function is said to be **monotonically increasing** if $f(x + \epsilon) \geq f(x)$ whenever $\epsilon > 0$, or **monotonically decreasing** if $f(x + \epsilon) \leq f(x)$ whenever $\epsilon > 0$. In either case, the function is said to be **monotonic**. A **periodic**

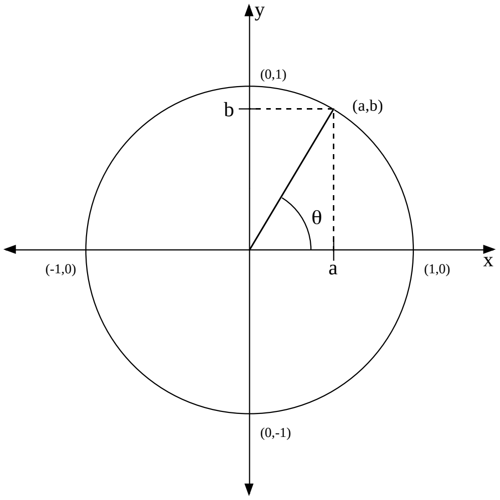

Figure 2-1: The unit circle

function is one for which $f(x+np) = f(x)$ for every integer n and some constant p. The **period** of the function is p.

2.2.6 Common functions

A **polynomial** is a function that matches a number to some combination of powers of that number. The **order of a polynomial** is the highest power that appears in that polynomial. For example, the function $f(x)$ given by

$$f(x) = x^3 + 6x^2 - 2x - 9 \tag{2.84}$$

is a third-order polynomial.

A **rational function** is a ratio of two polynomials. An example is

$$f(x) = \frac{x^4 - x^2 + 5}{x^3 + 7x - 2} \tag{2.85}$$

The **trigonometric functions** include **cosine**, **sine**, and **tangent**, and are defined in terms of right triangles or the unit circle (a circle of radius 1 centered at the origin), which appears in figure 2-1. Let θ be the angle measured counterclockwise from the x axis to a line segment with one endpoint at the origin $(0,0)$, as shown in the figure. This line segment intersects the unit circle at some point (a, b). The cosine of θ, written $\cos(\theta)$, is simply a, while the sine of θ, written $\sin(\theta)$, is b. The tangent of θ, written $\tan(\theta)$, is b/a. Note that $-1 \leq \cos(\theta) \leq 1$ and $-1 \leq \sin(\theta) \leq 1$ for all θ.

The cosine and sine of θ give the lengths of two sides of a right triangle. The hypoteneuse of the triangle goes from the origin to the circle, so it always has length 1. By the Pythagorean Theorem, the square of the length of the hypoteneuse of a right triangle is equal to the sum of the squares of the lengths of the other two sides. This means that, for every θ,

$$\cos^2(\theta) + \sin^2(\theta) = 1 \tag{2.86}$$

The angle θ is typically given, not in degrees, but in **radians**. There are 2π radians in $360°$. The trigonometric functions are periodic with period 2π, because any angle $\theta + n(2\pi)$ is in the same direction as θ; it just involves n extra rotations around the origin.

The **factorial** is a function defined only for positive integers. For any positive integer n, n factorial is written as $n!$ and is defined as follows.

$$n! = n \times (n-1) \times (n-2) \cdots \times 2 \times 1 \tag{2.87}$$

Note that

$$\frac{n!}{n} = (n-1)! \tag{2.88}$$

The **exponential function** is the value of a special number e raised to any power. It may be written as a polynomial.

$$e^a = 1 + a + \frac{1}{2!}a^2 + \frac{1}{3!}a^3 + \dots \tag{2.89}$$

This polynomial expression allows the exponential of a to be evaluated even when a is not an integer. The number e is called Euler's constant; its value is given by the following limit.

$$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n \approx 2.7183 \tag{2.90}$$

The exponential of a matrix, written $e^{\mathbf{A}}$, is itself a matrix, and may also be written as a polynomial.

$$e^{\mathbf{A}} = \mathbf{I} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \ldots \tag{2.91}$$

One important property of the matrix exponential is that it commutes with its exponent. This may be shown most easily by its polynomial representation.

$$\begin{aligned}\mathbf{A}e^{\mathbf{A}} &= \mathbf{AI} + \mathbf{AA} + \mathbf{A}\frac{1}{2!}\mathbf{A}^2 + \mathbf{A}\frac{1}{3!}\mathbf{A}^3 + \ldots \\ &= \mathbf{IA} + \mathbf{AA} + \frac{1}{2!}\mathbf{A}^2\mathbf{A} + \frac{1}{3!}\mathbf{A}^3\mathbf{A} + \ldots \\ &= e^{\mathbf{A}}\mathbf{A}\end{aligned} \tag{2.92}$$

Diagonalization of a matrix, or transformation to its Jordan form, also simplifies calculation of the exponential of that matrix. Let a matrix $\mathbf{A}$ have a complete set of eigenvectors $\mathbf{S}$, so that $\mathbf{A} = \mathbf{S\Lambda S}^{-1}$. Then

$$\begin{aligned}e^{\mathbf{A}} &= e^{\mathbf{S\Lambda S}^{-1}} \\ &= \mathbf{I} + \mathbf{S}^{-1}\mathbf{\Lambda S} + \frac{1}{2!}\left(\mathbf{S\Lambda S}^{-1}\right)^2 + \frac{1}{3!}\left(\mathbf{S\Lambda S}^{-1}\right)^3 + \ldots \\ &= \mathbf{S}\left(\mathbf{I} + \mathbf{\Lambda} + \frac{1}{2!}\mathbf{\Lambda}^2 + \frac{1}{3!}\mathbf{\Lambda}^3 + \ldots\right)\mathbf{S}^{-1} \\ &= \mathbf{S}e^{\mathbf{\Lambda}}\mathbf{S}^{-1}\end{aligned} \tag{2.93}$$

2.2.7 Complex numbers

The **imaginary number** i is defined as the square root of -1.

$$i = \sqrt{-1} \quad \Leftrightarrow \quad i^2 = -1 \tag{2.94}$$

A **complex number** is any sum of a real number and a real multiple of i. If some number x is a member of the set of all complex numbers (usually written as simply $x \in \mathbb{C}$), it has the form

$$x = a + bi \tag{2.95}$$

where a is called the **real part**, and bi is called the **complex part**.

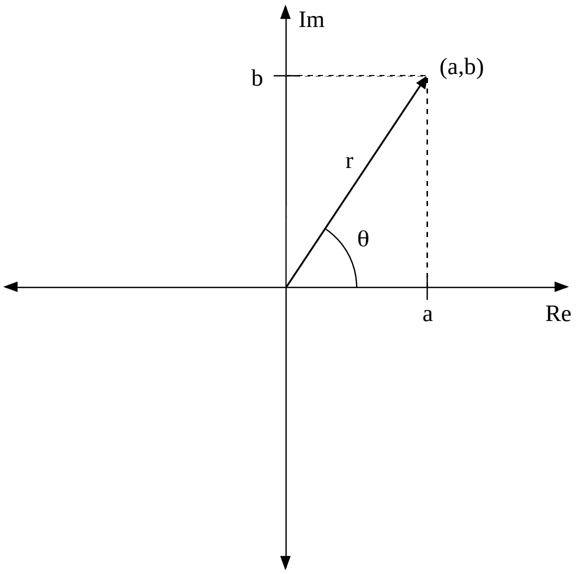

Figure 2-2: A number $a + bi$ in the complex plane

Addition and multiplication of two complex numbers $x = a + bi$ and $y = c + di$ are given by

$$\begin{aligned} x + y &= (a + b) + (c + d)i \\ x \times y &= (ac - bd) + (ad + bc)i \end{aligned} \tag{2.96}$$

It is fairly easy to show that $\langle \mathbb{C}, +, \times \rangle$ is a field.

The magnitude of a complex number x is the analogue of the absolute value of a real number, and is given by

$$|x| = \sqrt{a^2 + b^2} \tag{2.97}$$

The set of all complex numbers $\mathbb{C}$ is often thought of as a plane, with the x axis giving the real part and the y axis giving the complex part of each complex number. This is called the **complex plane**, and the number $a + bi$ is shown in the complex plane in figure 2-2.

Another way of expressing a complex number x is in terms of its **magnitude** r and **phase angle** θ, again as shown in figure 2-2. The magnitude and phase angle (often called simply the **phase**) are related to the real and complex parts by

$$\begin{aligned} r &= \sqrt{a^2+b^2} \\ \theta &= \cos^{-1}\left(\frac{a}{\sqrt{a^2+b^2}}\right) \\ &= \sin^{-1}\left(\frac{b}{\sqrt{a^2+b^2}}\right) \end{aligned} \tag{2.98}$$

while the real and complex parts are given in terms of the magnitude and phase by

$$\begin{aligned} a &= r\cos(\theta) \\ b &= r\sin(\theta) \end{aligned} \tag{2.99}$$

The square root of a complex number is most easily calculated using the magnitude and phase.

$$\sqrt{a+bi} = \sqrt{r\left(\cos(\theta)+i\sin(\theta)\right)} = \sqrt{r}\left(\cos\left(\frac{\theta}{2}\right)+i\sin\left(\frac{\theta}{2}\right)\right) \tag{2.100}$$

A complex number $a+bi$ for which $a=0$ is called **pure imaginary** or **strictly imaginary**. The exponential of a pure imaginary number is given by **Euler's formula**.

$$e^{ib} = \cos(b) + i\sin(b) \tag{2.101}$$

The exponential of a pure imaginary number is itself a complex number. Its real part is $\cos(b)$, and its imaginary part is $\sin(b)$. Note that the magnitude of the exponential of a pure imaginary number is always one.

$$|e^{ib}| = \sqrt{\cos^2(b)+\sin^2(b)} = 1 \tag{2.102}$$

The exponential of a pure imaginary number therefore lies on a circle of radius 1 centered at the origin of the complex plane.

2.3 Differential and Algebraic Equations

The unknowns in a system of algebraic equations are variables that take on a single value, so "solving the system" means calculating these unknown values. In differential equations, the unknowns are functions, rather than values, which introduces new issues that are not encountered in linear algebraic systems. For example, infinitely many functions satisfy a set of differential equations, so initial conditions are required in order to determine a unique solution. Coupling algebraic and differential equations adds yet another set of issues.

2.3.1 Differentiation and integration

Unless otherwise stated, the domain of a function will be assumed to be the real numbers, or an interval of the real numbers. Whenever $f(t)$ is written, it will be assumed that $t \in \mathbb{R}$.

A function is said to be **continuous at** t if

$$\lim_{\epsilon \to 0} \left(u(t+\epsilon) - u(t-\epsilon)\right) = 0 \tag{2.103}$$

If this is true for all t, than the function is simply called **continuous**.

Given a function $u(t)$, the **derivative of u** is the instantaneous rate of change of $u(t)$ at a particular value of t, and is denoted by $\frac{du}{dt}$, or u', or $\dot{u}$.

$$\frac{du}{dt} = u' = \dot{u} = \lim_{h \to 0} \frac{u(t+h) - u(t)}{h} \tag{2.104}$$

Note that $\dot{u}(t)$ is itself a function, and so it may have a derivative itself. The derivative of the derivative of u is the **second derivative of** u, and so forth. Repeated derivatives may be denoted in several ways; for example, the third derivative of u may be written using any of the following equivalent notations.

$$\left(\frac{d}{dt}\right)^3 u = u''' = \dddot{u} = u^{(3)} \tag{2.105}$$

The derivative of u is only defined at a particular value of t if the limit is the same regardless of which side of $u(t)$ the difference is taken; in other words, it must be true that

$$\lim_{h \to 0} \frac{u(t+h) - u(t)}{h} = \lim_{h \to 0} \frac{u(t) - u(t-h)}{h} \tag{2.106}$$

for the derivative to exist at $u(t)$.

C^i is the set of all functions $u(t)$ for which the i^{th} derivative exists for all t. If $\ddot{u}(t)$ is uniquely defined for all t, then $u \in C^2$. This may be restricted to a particular interval in t, so if the second derivative of u exists only for $t \in I$, then u is an element of C^2 on I. A common shorthand notation is "u is C^2 on I". A function that is C^0 is continuous.

There are two types of integrals, definite and indefinite. The **indefinite integral of u** is in some sense the inverse of the derivative.

$$\frac{d}{dt}\int u(t)dt = u(t) \tag{2.107}$$

A function $U(t)$ that is an indefinite integral of $u(t)$

$$U(t) = \int u(t)dt \tag{2.108}$$

is called an **antiderivative** of u, because $U'(t) = u(t)$, which may be shown by simply differentiating both sides of the definition of $U(t)$.

$$\begin{aligned} \frac{dU}{dt} &= \frac{d}{dt}\int u(t)dt \\ U'(t) &= u(t) \end{aligned} \tag{2.109}$$

The indefinite integral, like the derivative, is itself a function of t.

While the derivative of the integral of u is simply u, the reverse is not true. This is because by definition (2.104) the derivative of a constant is zero. Therefore, adding any constant c to $U(t)$ produces another antiderivative of u.

$$\frac{d}{dt}\left(U(t) + c\right) = U'(t) + 0 = U'(t) \tag{2.110}$$

The **definite integral** is a function of an *interval* of t, and is defined as follows. Suppose $U(t)$ is an antiderivative of $u(t)$, and t is further restricted to an interval of $\mathbb{R}$ given by $a \leq t \leq b$. Then the definite integral of $u(t)$ from $t = a$ to $t = b$ is defined as

$$\int_a^b u(t)dt = U(b) - U(a) \tag{2.111}$$

This definition is called the **Fundamental Theorem of Calculus**.

If the derivative of a function is known, the function itself is known up to a constant of integration, and may be expressed as either a definite or indefinite integral. If

$$\dot{u} = f(t) \tag{2.112}$$

then, by the Fundamental Theorem of Calculus,

$$u(t) = \int_0^t f(\tau)d\tau + u(0) \tag{2.113}$$

It is important to contrast an equation involving the derivative of a function (2.112) with a linear algebraic equation (2.3). The solution to a linear algebraic equation is a *constant*, and if a solution exists, it is unique. The solution to an equation like the one above (2.112) is a *function*, and if a solution exists, there are infinitely many other solutions as well. All of these solutions that satisfy the equation differ by an arbitrary constant, which here is $u(0)$.

2.3.2 Rules of differentiation

There are three simple rules of differentiation that arise again and again. The first is called the **chain rule**. It gives the derivative of a function that is a composition of two or more functions.

$$\frac{d}{dt}\left(f(g(t))\right) = f'(g(t))g'(t) \tag{2.114}$$

The second is the **product rule** for differentiating the product of two functions.

$$\frac{d}{dt}\left(f(t)g(t)\right) = f'(t)g(t) + f(t)g'(t) \tag{2.115}$$

Third is the **power rule** for differentiating a number raised to a power.

$$\frac{d}{dt}t^n = nt^{n-1} \tag{2.116}$$

The power rule is particularly useful for differentiating polynomials. For example,

$$\frac{d}{dt}\left(x^3 + 4x^2 + 9x\right) = 3x^2 + 8x + 9 \tag{2.117}$$

Recall that the exponential function is a polynomial. The power rule allows its derivative to be calculated very easily.

$$\begin{aligned}\frac{d}{dt}e^t &= \frac{d}{dt}\left(1 + t + \frac{1}{2!}t^2 + \frac{1}{3!}t^3 + \dots\right) \\ &= \left(0 + 1 + \frac{2}{2!}t + \frac{3}{3!}t^2 + \dots\right) \\ &= \left(0 + 1 + \frac{1}{1!}t + \frac{1}{2!}t^2 + \dots\right) \\ &= e^t\end{aligned} \tag{2.118}$$

So, the exponential function is its own derivative!

2.3.3 Norms of functions

For a function of one variable, say $u(t)$, defined[4] on some interval $a \leq t \leq b$, the L^2 **norm of** u is defined as follows.

$$||u|| = \left(\int_a^b u^2 dt\right)^{\frac{1}{2}} \tag{2.119}$$

The interval may also be infinite, in which case $a = -\infty$ and $b = \infty$. For a vector of functions $\mathbf{u}(t)$, the L^2 norm is defined as

$$||\mathbf{u}|| = \left(\int_a^b (\mathbf{u} \cdot \mathbf{u}) dt\right)^{\frac{1}{2}} \tag{2.120}$$

For a function of several variables, say $u(t, x)$, on the domain $\Omega = \{(t, x) : a \leq t \leq b, c \leq x \leq d\}$, $||u(t, \cdot)||$ is defined as

$$||u(t, \cdot)|| = \left(\int_c^d u^2(t, x) dx\right)^{\frac{1}{2}} \tag{2.121}$$

while

$$||u(\cdot)|| = \left(\int_a^b \int_c^d u^2(t, x) dx dt\right)^{\frac{1}{2}} \tag{2.122}$$

[4] The value of $u(t)$ is assumed to be real, not complex, for all t. If $u(t)$ is allowed to be complex, then the norms must be defined slightly differently.

This is sometimes written using a more compact notation. Let

$$\mathbf{z} = \begin{bmatrix} t \\ x \end{bmatrix} \tag{2.123}$$

Then the same norm may be written as

$$||u(\cdot)|| = \left(\int_{\Omega} u^2 d\mathbf{z} \right)^{\frac{1}{2}} \tag{2.124}$$

Using the same notation, the norm of a vector of functions that depend on multiple independent variables $\mathbf{u}(\mathbf{z})$ over a domain Ω is

$$||\mathbf{u}(\cdot)|| = \left(\int_{\Omega} (\mathbf{u} \cdot \mathbf{u}) d\mathbf{z} \right)^{\frac{1}{2}} \tag{2.125}$$

2.3.4 Scalar ordinary differential equations

Differential equations involve both u and its derivatives. Possibly the simplest ordinary differential equation, or **ODE**, is

$$\dot{u} + au = 0 \tag{2.126}$$

where $a \in \mathbb{R}$ is a constant. The equation is called **homogeneous** because the right-hand side is identically zero; if it were instead some function of t, it would be called **inhomogeneous**, and $f(t)$ would be called the **forcing function**.

This equation may be solved immediately by inspection. Recall that the exponential function is its own derivative. Using the chain rule,

$$\frac{d}{dt} e^{kt} = ke^{kt} \tag{2.127}$$

so any function $u(t)$ of the form

$$u(t) = c_0 e^{-at} \tag{2.128}$$

where c_0 is an arbitrary constant, will satisfy the equation. An **initial condition** determines a unique solution from this family of solutions. If

$$u(0) = u_0 \tag{2.129}$$

then the unique solution that passes through $(u, t) = (u_0, 0)$ and satisfies the original ordinary differential equation is

$$u(t) = u_0 e^{-at} \tag{2.130}$$

The solution to a homogeneous differential equation is a **homogeneous solution**.

The solution of the related inhomogeneous equation

$$\dot{u} + au = f(t) \tag{2.131}$$

is given by

$$u(t) = u_0 e^{-at} + e^{-at} \int_0^t e^{a\tau} f(\tau) d\tau \tag{2.132}$$

The second term on the righthand side matches the forcing function and is called the **particular solution**. The solution to an inhomogeneous differential equation will consist of a homogeneous solution and a particular solution.

Another way of looking at the solution of an inhomogeneous equation of this form (at a particular value of t) is as the solution of the related homogeneous equation, but with a different initial condition. Let

$$u_0^* = \int_0^t e^{a\tau} f(\tau) d\tau \tag{2.133}$$

Then

$$u(t) = (u_0 + u_0^*) e^{-at} \tag{2.134}$$

This is called **Duhamel's principle**. Note that u_0^* is different at different values of t.

For a more general differential equation of the form

$$\dot{u} = f(u, t) \tag{2.135}$$

a differentiable solution is guaranteed to exist and be unique if $f(u, t)$ is **Lipschitz continuous** [37]. A function $f(u, t)$ is Lipschitz continuous iff

$$|f(u_1, t) - f(u_2, t)| \leq L|u_1 - u_2| \tag{2.136}$$

for some finite scalar L.

2.3.5 Systems of ordinary differential equations

A system of ordinary differential equations is itself often called an ODE. Consider a homogeneous ODE the form

$$\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} = \mathbf{0} \tag{2.137}$$

This system is called **linear**, because $\mathbf{A}$ is constant.

The solution is given by

$$\mathbf{u}(t) = e^{-\mathbf{A}t}\mathbf{u}_0 \tag{2.138}$$

if the initial conditions are

$$\mathbf{u}(0) = \mathbf{u}_0 \tag{2.139}$$

Clearly, the number of initial conditions must equal the number of dependent variables.

The solution to the related inhomogeneous linear system

$$\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} = \mathbf{f}(t) \tag{2.140}$$

for the same initial conditions is given by

$$\mathbf{u}(t) = e^{-\mathbf{A}t}\mathbf{u}_0 + e^{-\mathbf{A}t}\int_0^t e^{\mathbf{A}\tau}\mathbf{f}(\tau)d\tau \tag{2.141}$$

2.3.6 Consistent initialization

Determination of a unique solution to a linear ODE system requires specification of n initial conditions, which often consist of the n values of $\mathbf{u}$ at time $t = 0$. Sometimes, however, one may wish to specify values of $\dot{\mathbf{u}}$ at $t = 0$; for example, specifying $\dot{\mathbf{u}}(0) = \mathbf{0}$ means that the system is starting from a steady state.

Consistent initial conditions are initial conditions that uniquely determine the solution to a system of differential equations. For a linear ODE, a unique solution $\mathbf{u}(t)$ is determined when all values of $\dot{\mathbf{u}}(0)$ and $\mathbf{u}(0)$ are known.

The problems of finding $\mathbf{u}(t)$ and consistent values of $\dot{\mathbf{u}}(0)$ and $\mathbf{u}(0)$ differ in fundamental ways[5]. The former problem involves finding a family of *functions of* t that satisfy the ODE system, and was the subject of the previous section. The latter problem involves finding a set of *values* of the functions and their derivatives at a particular value of t, and is an algebraic problem like those covered in the first part of this chapter. Another significant difference is that, in the ODE problem, there are n unknowns, which are the functions $\mathbf{u}(t)$, while in the consistent initialization problem, there are $2n$ unknowns, which are the values of $\mathbf{u}(0)$ and $\dot{\mathbf{u}}(0)$.

Suppose the equations are homogeneous and the initial conditions are $\mathbf{u}(0) = \mathbf{b}$. Once these are provided, the ODE gives the values of $\dot{\mathbf{u}}(0)$. The overall consistent initialization problem is

$$\begin{aligned} \dot{\mathbf{u}}(0) + \mathbf{A}\mathbf{u}(0) &= \mathbf{0} \\ \mathbf{u}(0) &= \mathbf{b} \end{aligned} \tag{2.142}$$

which has the solution

$$\begin{aligned} \mathbf{u}(0) &= \mathbf{b} \\ \dot{\mathbf{u}}(0) &= -\mathbf{A}\mathbf{b} \end{aligned} \tag{2.143}$$

This system has a unique solution, so the initial conditions are consistent.

Now, suppose that the initial conditions are instead $\dot{\mathbf{u}}(0) = \mathbf{b}$. The ODE must then be used to determine the values of $\mathbf{u}(0)$. The overall consistent initialization problem is now

$$\begin{aligned} \dot{\mathbf{u}}(0) + \mathbf{A}\mathbf{u}(0) &= \mathbf{0} \\ \dot{\mathbf{u}}(0) &= \mathbf{b} \end{aligned} \tag{2.144}$$

and the solution

$$\begin{aligned} \mathbf{u}(0) &= -\mathbf{A}^{-1}\mathbf{b} \\ \dot{\mathbf{u}}(0) &= \mathbf{b} \end{aligned} \tag{2.145}$$

[5] Unfortunately, the ODE problem and its associated consistent initialization problem are often written using identical notation, where both the unknown functions $\mathbf{u}(t)$ and their derivatives $\dot{\mathbf{u}}(t)$, and the unknown values $\mathbf{u}(0)$ and $\dot{\mathbf{u}}(0)$, are denoted simply as $\mathbf{u}$ and $\dot{\mathbf{u}}$.

exists and is unique iff **A** is invertible. If **A** does not have an inverse, then the initial conditions do not uniquely determine all $\mathbf{u}(0)$ and therefore are not consistent.

Because obtaining $\dot{\mathbf{u}}(0)$ from $\mathbf{u}(0)$ for an ODE is always possible, one is typically concerned only with obtaining $\mathbf{u}(0)$. However, many numerical integrators require a consistent $\dot{\mathbf{u}}(0)$ to start efficiently. For this reason, $\dot{\mathbf{u}}(0)$ is sometimes considered to be a purely a numerical consideration. For an ODE, once $\mathbf{u}(0)$ is known, $\dot{\mathbf{u}}(0)$ is always uniquely determined.

2.3.7 Differential-algebraic systems

Consider a system of the form

$$\mathbf{A}\dot{\mathbf{u}} + \mathbf{B}\mathbf{u} = \mathbf{0} \tag{2.146}$$

If **A** is invertible, multiplication on the left by $\mathbf{A}^{-1}$ produces an ODE. The solution may be found as in the previous section.

If, however, **A** is singular, this is a mixed system of **differential-algebraic equations**, or a **DAE**. Because **A** and **B** are constant matrices, the DAE (2.146) is called **linear time invariant**. If **A** or **B** instead vary with t, the DAE is called **linear time varying**. The most general DAE is a system of nonlinear functions of $\dot{\mathbf{u}}$ and $\mathbf{u}$, and is called a **nonlinear** DAE:

$$\mathbf{f}(\dot{\mathbf{u}}, \mathbf{u}, t) = \mathbf{0} \tag{2.147}$$

DAEs have many properties for which there is no analogy among ODEs. It is possible for a DAE to have a unique solution before specification of any initial conditions. Sometimes specifying values for $\mathbf{u}(0)$ as initial conditions may be inconsistent with a particular DAE, and thus no solution can satisfy both the equations and those initial conditions. Existence of a solution to an inhomogeneous DAE may require existence of derivatives of the forcing functions.

A linear time invariant DAE is called **solvable** iff the coefficient matrix pair $(\mathbf{A}, \mathbf{B})$ forms a regular pencil and the forcing functions are sufficiently differentiable [8]. Solvability is a necessary condition for existence and uniqueness of a solution.

Further conditions involving consistency of the initial conditions must be met in order to guarantee existence and uniqueness of the solution to a solvable system.

The general form of a solution to an (assumed solvable) homogeneous, linear time invariant DAE may be constructed as follows [10]. First, multiply the system on the left by the inverse of any invertible member of the coefficient matrix pencil. For example, let λ be some scalar for which $(\mathbf{B}+\lambda\mathbf{A})$ is invertible. Let $\hat{\mathbf{A}}_\lambda = (\mathbf{B}+\lambda\mathbf{A})^{-1}\mathbf{A}$ and $\hat{\mathbf{B}}_\lambda = (\mathbf{B}+\lambda\mathbf{A})^{-1}\mathbf{B}$. In general two arbitrary matrices $\mathbf{A}$ and $\mathbf{B}$ do not commute, but $\hat{\mathbf{A}}_\lambda$ and $\hat{\mathbf{B}}_\lambda$ commute; that is, $\hat{\mathbf{A}}_\lambda\hat{\mathbf{B}}_\lambda = \hat{\mathbf{B}}_\lambda\hat{\mathbf{A}}_\lambda$.

Dropping the subscript λ, the solution to the DAE (2.146) is given by

$$\mathbf{u}(t) = e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D\mathbf{u}_0 \tag{2.148}$$

where $\mathbf{u}_0$ is a set of consistent initial conditions.

For an *inhomogeneous* linear time invariant DAE, the solution is the sum of the homogeneous solution (2.148) and a particular solution, and is given by

$$\mathbf{u}(t) = e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\int_0^t e^{\hat{\mathbf{A}}^D\hat{\mathbf{B}}\tau}\hat{\mathbf{A}}^D\hat{\mathbf{f}}(\tau)d\tau + \left(\mathbf{I} - \hat{\mathbf{A}}\hat{\mathbf{A}}^D\right)\sum_{i=0}^{k-1}(-1)^i\left(\hat{\mathbf{A}}\hat{\mathbf{B}}^D\right)^i\hat{\mathbf{B}}^D\hat{\mathbf{f}}^{(i)}(t) \tag{2.149}$$

where k is the nilpotency of $\mathbf{N}$ in the Jordan form of $\hat{\mathbf{A}}$.

Note that the inhomogeneous solution depends on some elements of $\hat{\mathbf{f}}^{(k-1)}(t)$, the $(k-1)^{th}$ derivative of the forcing functions. Wherever particular elements of $\hat{\mathbf{f}}(t)$ are not sufficiently differentiable, the solution does not exist.

The subscript λ was dropped because these expressions for the solution (2.148 - 2.149) are independent of the particular value of λ chosen. If μ is some other scalar for which $(\mathbf{B}+\mu\mathbf{A})$ is invertible, the following properties hold.

$$\begin{aligned}\hat{\mathbf{A}}_\lambda\hat{\mathbf{A}}_\lambda^D &= \hat{\mathbf{A}}_\mu\hat{\mathbf{A}}_\mu^D\\ \hat{\mathbf{A}}_\lambda^D\hat{\mathbf{B}}_\lambda &= \hat{\mathbf{A}}_\mu^D\hat{\mathbf{B}}_\mu\\ \hat{\mathbf{A}}_\lambda^D\hat{\mathbf{f}}_\lambda &= \hat{\mathbf{A}}_\mu^D\hat{\mathbf{f}}_\mu\\ \hat{\mathbf{B}}_\lambda^D\hat{\mathbf{f}}_\lambda &= \hat{\mathbf{B}}_\mu^D\hat{\mathbf{f}}_\mu\end{aligned} \tag{2.150}$$

The solution may also be constructed using a change of variables [30, 89]. Again assume that the DAE is solvable, which implies that the coefficient matrices form a regular pencil. Let **P** and **Q** be the invertible matrices that take the coefficient matrix pair to its Weierstrass canonical form. Multiplying the system on the left by **P** and introducing new variables $\mathbf{v} = \mathbf{Q}^{-1}\mathbf{u}$ and forcing terms $\mathbf{g} = \mathbf{P}\mathbf{f}$ produces a system of the form

$$\begin{bmatrix} \mathbf{I} & \\ & \mathbf{N} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_1 \\ \dot{\mathbf{v}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{J} & \\ & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1(t) \\ \mathbf{g}_2(t) \end{bmatrix} \tag{2.151}$$

The two block rows are decoupled and so may be solved independently. The first block row is called the **differential subsystem**. It is an ODE of the form considered earlier (2.140), and so has a unique solution for any initial condition $\mathbf{v}_1(0) = \mathbf{v}_{10}$.

The second block row is the **algebraic subsystem**. If **N** is nonzero, it will contain differential equations, but it is nevertheless equivalent to a system of algebraic equations. To see this, write the algebraic subsystem as

$$\left(\mathbf{N}\frac{d}{dt} + \mathbf{I}\right)\mathbf{v}_2 = \mathbf{g}_2(t) \tag{2.152}$$

and let

$$\left(\mathbf{N}\frac{d}{dt} + \mathbf{I}\right)^* = \mathbf{I} + \sum_{i=1}^{k-1}(-1)^i\mathbf{N}^i\left(\frac{d}{dt}\right)^i \tag{2.153}$$

where k is the nilpotency of **N**. Applying this operator to the algebraic subsystem gives $\mathbf{v}_2$ as a unique function of the forcing functions and their derivatives; no arbitrary constants appear in the solution.

$$\left(\mathbf{N}\frac{d}{dt} + \mathbf{I}\right)^*\left(\mathbf{N}\frac{d}{dt} + \mathbf{I}\right)\mathbf{v}_2 = \mathbf{v}_2 = \left(\mathbf{N}\frac{d}{dt} + \mathbf{I}\right)^*\mathbf{g}_2(t) \tag{2.154}$$

From the definition of $\left(\mathbf{N}\frac{d}{dt} + \mathbf{I}\right)^*$, it is clear that $\mathbf{v}_2(t)$ depends on up to $k-1$ derivatives of the forcing functions $\mathbf{g}_2$. Wherever these derivatives fail to exist, $\mathbf{v}_2$ will also fail to exist.

2.3.8 The index of a linear DAE

The **index**, typically denoted ν, of a solvable, linear time invariant DAE is defined as equal to the nilpotency k of **N** in the Weierstrass canonical form of the coefficient matrix pair. A DAE of index zero is an ODE. A DAE of index 2 or greater is considered to be **high index**.

What is the significance of having a high index? High index DAEs can have hidden algebraic relationships between the dependent variables and/or their derivatives. This complicates the problem of providing proper initial conditions for the system, and also of integrating the problem numerically[6].

For a linear ODE (which is an index-0 DAE) that consists of n differential equations, specification of all n values of $\mathbf{u}(0)$ always produces a solvable consistent initialization problem. For a DAE, let p be the number of differential equations or differential variables, whichever is less[7]. For an index-1 DAE, p initial conditions are typically required to produce a solvable consistent initialization problem. For a high index DAE, however, fewer than p initial conditions must be given. In fact, it is possible that no initial conditions may be arbitrarily specified for a high index DAE.

For example, consider a **derivative chain** of length 3 [8].

$$\begin{aligned} \dot{u}_1 + u_2 &= f_1(t) \\ \dot{u}_2 + u_3 &= f_2(t) \\ u_1 &= f_3(t) \end{aligned} \tag{2.155}$$

The index of this system is 3, so it is high index. Although there are two differential equations, the solution is algebraic.

$$\begin{aligned} u_1 &= f_3(t) \\ u_2 &= f_1(t) - \dot{f}_3(t) \\ u_3 &= f_2(t) - \dot{f}_1(t) + \ddot{f}_3(t) \end{aligned} \tag{2.156}$$

[6] See Petzold [69] or Sincovec et al [78] for an exploration of the issues surrounding numerical solution of high index DAEs. Loosely speaking, if the system is low index, standard methods for stiff ODEs may be applied. If the system is high index, such codes can only be applied in special cases and with great caution.

[7] Here a differential equation is one that contains a derivative term, and a differential variable is one for which a derivative appears in one or more equation. In general these may be different from the differential and algebraic subsystems defined in the previous section - note that there may be differential equations in the algebraic subsystem, for example.

Although $p = 2$, no arbitrary constants appear anywhere in the solution, so no initial conditions may be specified.

An interpretation of a high index DAE is that it is a system that contains implicit constraints on the derivatives of the variables. These implicit constraints take up some of the degrees of freedom in the consistent initialization problem. For the derivative chain above, the initialization problem starts with the equations themselves.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\mathbf{u}}(0) + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{u}(0) = \mathbf{f}(t) \tag{2.157}$$

There are two differential variables, so it is reasonable to expect that two initial conditions are required in order to determine $\dot{u}_1(0)$, $u_1(0)$, $\dot{u}_2(0)$, $u_2(0)$, and $u_3(0)$.

However, differentiating the third equation produces a new equation independent of the first three.

$$\dot{u}_1(0) = \dot{f}_3(0) \tag{2.158}$$

This equation takes the place of one arbitrarily specified initial condition in the consistent initialization problem.

Differentiating the first equation, and differentiating the third equation a second time, produces two new equations in only one new variable ($\ddot{u}_1(0)$).

$$\begin{aligned} \ddot{u}_1(0) + \dot{u}_2(0) &= \dot{f}_1(0) \\ \ddot{u}_1(0) &= \ddot{f}_3(0) \end{aligned} \tag{2.159}$$

Taken together, these two new equations in the one new variable $\ddot{u}_1(0)$ form a second constraint, that was again "hidden" or implicit in the original equations. After using one of these equations to eliminate the new variable from the other, the resulting constraint takes the place of another equation in the consistent initialization problem, for a total of five equations in the five unknowns $\dot{u}_1(0)$, $u_1(0)$, $\dot{u}_2(0)$, $u_2(0)$, and $u_3(0)$. These five equations are nonsingular, so no initial conditions may be specified arbitrarily.

Suppose that an initial condition, perhaps $u_1(0) = k$, had been arbitrarily specified. Because $u_1(0)$ must equal $f_3(0)$ in order to satisfy the equations, no solution can satisfy both the initial condition on $u_1(0)$ and the equations for an arbitrary k. In other words, $u_1(0) = k$ is not a consistent initial condition. In general, variables $\mathbf{u}(0)$ or their derivatives $\dot{\mathbf{u}}(0)$ which *can* be assigned arbitrary values and still allow solution of the original system are called **dynamic degrees of freedom**. For linear time invariant DAEs, the number of dynamic degrees of freedom is equal to the dimension of the differential subsystem.

An ODE to which are appended a set of algebraic functions of the differential variables forms an index-1 DAE. Such a system has the form

$$\begin{aligned} \dot{\mathbf{u}} + \mathbf{B}\mathbf{u} &= \mathbf{f}_1(t) \\ \mathbf{C}\mathbf{y} + \mathbf{D}\mathbf{u} &= \mathbf{f}_2(t) \end{aligned} \tag{2.160}$$

with $\mathbf{C}$ invertible. Here $\mathbf{u}$ are called the **differential variables**, and are given as the solution to the ODE. $\mathbf{y}$ are called the **algebraic variables**, which are uniquely determined by the forcing function $\mathbf{f}_2(t)$ and the values of the differential variables.

Some linear index-1 DAEs cannot be written in such a form. For example, the following system [63] is index-1.

$$\begin{aligned} \dot{u}_1 + \dot{u}_2 &= f_1(t) \\ u_1 + 3u_2 &= f_2(t) \end{aligned} \tag{2.161}$$

Although it is index-1, this system has an implicit constraint found by differentiating the second equation. This constraint and the original equations together comprise three equations in the four unknowns $\dot{u}_1(0)$, $u_1(0)$, $\dot{u}_2(0)$, and $u_2(0)$, so only one initial condition is needed. Systems of index 1 that have one or more implicit constraints on the differential variables are sometimes called **special index-1 systems**.

2.3.9 Nonlinear DAEs and the derivative array equations

For nonlinear DAEs,

$$\mathbf{f}(\dot{\mathbf{u}}, \mathbf{u}, t) = \mathbf{0} \tag{2.162}$$

the index can no longer be defined in terms of coefficient matrices. Different approaches have been taken [11]. The **differentiation index** is defined as the minimum number of times some or all of the equations must be differentiated in order to uniquely determine $\dot{\mathbf{u}}$ as a continuous function of $\mathbf{u}$ and t.

Writing $(\frac{d}{dt})^i\mathbf{f}(\dot{\mathbf{u}}, \mathbf{u}, t)$ as $\mathbf{f}_{[i]}(\dot{\mathbf{u}}, \mathbf{u}, t)$ and defining

$$\mathbf{u}_{[i]} = \begin{bmatrix} \frac{d}{dt}\mathbf{u} \\ \left(\frac{d}{dt}\right)^2 \mathbf{u} \\ \vdots \\ \left(\frac{d}{dt}\right)^i \mathbf{u} \end{bmatrix} \tag{2.163}$$

repeated differentiation of the DAE produces the following system of equations.

$$\begin{aligned} \mathbf{f}_{[0]}(\mathbf{u}_{[1]}, \mathbf{u}, t) &= \mathbf{0} \\ \mathbf{f}_{[1]}(\mathbf{u}_{[2]}, \mathbf{u}, t) &= \mathbf{0} \\ \mathbf{f}_{[2]}(\mathbf{u}_{[3]}, \mathbf{u}, t) &= \mathbf{0} \\ &\vdots \end{aligned} \tag{2.164}$$

Let the first $k+1$ block rows be written as

$$\mathcal{F}_{[k]}(\mathbf{u}_{[k+1]}, \mathbf{u}, t) = \mathbf{0} \tag{2.165}$$

These are the k^{th} **derivative array equations** [31]. The differentiation index ν_D is thus the smallest k such that $\mathcal{F}_{[k]}$ uniquely determines $\dot{\mathbf{u}}$ as a continuous function of $\mathbf{u}$ and t.

Linear time varying systems have the form

$$\mathbf{A}(t)\dot{\mathbf{u}} + \mathbf{B}(t)\mathbf{u} = \mathbf{f}(t) \tag{2.166}$$

For such systems, the derivative array equations are themselves a linear time varying system. For example, $\mathcal{F}_{[2]}$ is given by

$$\begin{bmatrix} \mathbf{A}^{(0)} & & \\ (\mathbf{A}^{(1)} + \mathbf{B}^{(0)}) & \mathbf{A}^{(0)} & \\ (\mathbf{A}^{(2)} + 2\mathbf{B}^{(1)}) & (2\mathbf{A}^{(1)} + \mathbf{B}^{(0)}) & \mathbf{A}^{(0)} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \ddot{\mathbf{u}} \\ \dddot{\mathbf{u}} \end{bmatrix} + \begin{bmatrix} \mathbf{B}^{(0)} \\ \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{f}^{(0)} \\ \mathbf{f}^{(1)} \\ \mathbf{f}^{(2)} \end{bmatrix} = \mathbf{0} \tag{2.167}$$

or more simply

$$\mathcal{A}_3\mathbf{u}_{[3]} = -\mathcal{B}_3\mathbf{u} + \mathbf{f}_{[3]} \tag{2.168}$$

Larger systems may be generated quickly by recursion. For the first column of a new row i, the $(i,1)^{th}$ element is the sum of $\mathbf{B}^{(i-2)}$ and the derivative of the $(i-1,j)^{th}$ element. The $(i,j)^{th}$ element of $\mathcal{A}_k$ is the sum of the $(i-1,j-1)^{th}$ element and the derivative of the $(i-1,j)^{th}$ element. Note that, for linear time invariant systems, $\mathbf{A}^{(i)} = \mathbf{B}^{(i)} = \mathbf{0}$ for $i \neq 0$.

Let $\mathbf{u} \in \mathbb{R}^n$. Then the matrix $\mathcal{A}_k$ is called **smoothly 1-full** if there is a smooth nonsingular $\mathbf{R}(t)$ such that

$$\mathbf{R}\mathcal{A}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \tag{2.169}$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$. The differentiation index ν_D is the smallest integer such that $\mathcal{A}_{\nu_D+1}$ is smoothly 1-full and has constant rank.

Now, consider the following simple nonlinear system.

$$\begin{aligned} \dot{u}_1 + u_2 &= f_1(t) \\ u_2^3 &= 0 \end{aligned} \tag{2.170}$$

This DAE consists of an ordinary differential equation involving the differential variable u_1, to which is appended an algebraic equation that uniquely defines the algebraic variable u_2. However, unlike linear systems of this description, the index is not 1.

The next three block rows of the derivative array equations are as follows.

$$\begin{aligned} \ddot{u}_1 + \dot{u}_2 &= f_1'(t) \\ 3u_2^2\dot{u}_2 &= 0 \end{aligned} \tag{2.171}$$

$$\begin{aligned} \dddot{u}_1 + \ddot{u}_2 &= f_1''(t) \\ 6u_2\dot{u}_2^2 + 3u_2^2\ddot{u}_2 &= 0 \end{aligned} \tag{2.172}$$

$$\begin{aligned} \ddddot{u}_1 + \dddot{u}_2 &= f_1'''(t) \\ 6\dot{u}_2^3 + 18u_2\dot{u}_2\ddot{u}_2 + 3u_2^2\dddot{u}_2 &= 0 \end{aligned} \tag{2.173}$$

The first equation in each new block row of the derivative array includes successively higher derivatives of u_1 and does not give $\dot{u}_2$ as a function of $\mathbf{u}$ and t only. Because

$u_2^3 = 0$, $u_2 = 0$ and the second equation in each block row is identically $0 = 0$ for one (2.171) and two (2.172) differentiations of the original system. Only after three differentiations is $\dot{u}_2$ given as a unique function of $\mathbf{u}$ and t (not surprisingly, it is identically zero). Because three differentiations were required, the index of this system is 3. This example demonstrates that a nonlinear system may have an arbitrarily high index, even when it is simply a fully determined ODE coupled to an algebraic equation that uniquely determines the algebraic variable.

The **perturbation index** [11] of the DAE (2.162) is defined as the smallest integer ν_P such that if

$$\mathbf{f}(\dot{\mathbf{v}}, \mathbf{v}, t) = \mathbf{g}(t) \tag{2.174}$$

for sufficiently smooth $\mathbf{g}$, then there is an estimate

$$||\mathbf{v}(t) - \mathbf{u}(t)|| \leq C(||\mathbf{v}(0) - \mathbf{u}(0)|| + ||\mathbf{g}||_{\nu_P - 1}^{T}) \tag{2.175}$$

for sufficiently small $\mathbf{g}$ and finite scalar C that may depend on t. The norm $||\mathbf{g}||_p^T$ is the norm of the first p derivatives of $\mathbf{g}$ over the interval $(0, T)$. More precisely, it is the sum of the maximum norm of $\mathbf{g}$, of $\mathbf{g}^{(1)}$, and of all successive derivatives up to maximum order p, over $(0, T)$.

$$||\mathbf{g}||_m^t = \sum_{i=0}^{m} \max_{t \in (0,T)} ||\mathbf{g}^{(i)}|| \tag{2.176}$$

The perturbation index is a property of the solution, rather than of the equations. The differentiation and perturbation index are equal for linear time invariant DAEs, but may differ for nonlinear problems. A system with a high differentiation index is one that has some implicit constraints. Just as with linear systems, these constraints reduce the number of initial conditions that must be specified in order to determine a unique solution.

Index analysis provides a wealth of information about the mathematical properties of a DAE; in particular, it gives the number of initial conditions required to determine a unique solution and the maximum order of derivatives of forcing functions that appear in the solution. Because the perturbation and differentiation indices are equal for linear systems, both are given by the index of nilpotency of **N** in the algebraic subsystem.

2.3.10 Automated index analysis

For nonlinear systems, and even for large linear systems, calculation of the index by transformation to the canonical form is impossible or impractical, respectively. Because the differentiation index is a property of the equations, while the perturbation index is a property of the analytical solution (which is usually unknown for dynamic flowsheet models), algorithms that allow a process simulator to attempt to perform index analysis on large or nonlinear systems are typically based on the differentiation index. In particular, Pantelides' algorithm [63], although designed to identify the number of dynamic degrees of freedom, has been employed successfully in dynamic process simulators to analyze the differentiation index of lumped flowsheet models.

Pantelides' algorithm works by identifying subsets of k equations, called **minimal structurally singular subsets**, that upon differentiation will produce fewer than k new variables. Here a "new variable" is meant in the context of a consistent initialization problem, where $\dot{u}(0)$ and $u(0)$ are considered to be distinct variables. The algorithm differentiates such a subset of equations and performs the analysis again, until no more minimal structurally singular subsets can be located.

The algorithm examines the structure of the system, which is given by the **incidence matrix**. The incidence matrix is determined simply by the occurrences of variables and their derivatives, and may be constructed very easily, even for nonlinear systems. Consider, for example, the derivative chain example (2.155). The incidence matrix for this system is

$$\begin{array}{r|ccccc|} & \dot{u}_1 & \dot{u}_2 & u_1 & u_2 & u_3 \\ \text{Equation 1} & \times & & & \times & \\ \text{Equation 2} & & \times & & & \times \\ \text{Equation 3} & & & \times & & \end{array} \tag{2.177}$$

The third equation may be differentiated without producing a new variable ($\dot{u}_1$ already appears in the first equation), so equation 3 forms the first minimal structurally singular subset. Differentiating it produces a system with the following incidence matrix.

$$\begin{array}{l|ccccc|} & \dot{u}_1 & \dot{u}_2 & u_1 & u_2 & u_3 \\ \text{Equation 1} & \times & & & \times & \\ \text{Equation 2} & & \times & & & \times \\ \text{Equation 3} & \times & & & & \end{array} \tag{2.178}$$

Now, the first and third equations together form a minimal structurally singular subset, because differentiation produces only one new variable ($\ddot{u}_1$). Differentiating these two equations and replacing $\dot{u}_1$ with $\ddot{u}_1$ again gives a new system with a new incidence matrix.

$$\begin{array}{l|ccccc|} & \ddot{u}_1 & \dot{u}_2 & u_1 & u_2 & u_3 \\ \text{Equation 1} & \times & \times & & & \\ \text{Equation 2} & & \times & & & \times \\ \text{Equation 3} & \times & & & & \end{array} \tag{2.179}$$

At this point, no more structurally singular subsets of equations exist. Every subset of k equations produces k new variables upon differentiation. Starting from the original three equations and five variables, the algorithm produced a total of three new equations through differentiation, along with one new variable ($\ddot{u}_1$), for a total of six equations in six unknowns. This means that no dynamic degrees of freedom exist for this system.

Because it works only with the occurrence of variables and their derivatives, this algorithm may be applied just as easily to nonlinear systems. As an example, consider the simple nonlinear system introduced earlier (2.170). The incidence matrix for this system is

$$\begin{array}{l|ccc|} & \dot{u}_1 & u_1 & u_2 \\ \text{Equation 1} & \times & & \times \\ \text{Equation 2} & & & \times \end{array} \tag{2.180}$$

No structurally singular subsets of equations exist in this system, so the algorithm terminates without performing any differentiations. The original two equations relate the three unknowns $\dot{u}_1$, u_1, and u_2, so there is one dynamic degree of freedom.

The information provided by Pantelides' algorithm, namely the number of times that each equation has been differentiated, has been used to estimate the differentiation index. If the derivative of every variable appears in the final system of equations produced by the algorithm, the index should equal the maximum number of times any equation was differentiated; otherwise, the index should be one greater than the maximum number of differentiations[8].

As noted in the original paper [63], systems with numerical singularity may not be differentiated a sufficient number of times. For example, consider the following simple, linear system.

$$\begin{aligned} \dot{u}_1 + \dot{u}_2 + u_1 &= 5 \\ \dot{u}_1 + \dot{u}_2 + u_2 &= 3 \end{aligned} \tag{2.181}$$

The incidence matrix for this system is

$$\begin{array}{c|cccc} & \dot{u}_1 & \dot{u}_2 & u_1 & u_2 \\ \hline \text{Equation 1} & \times & \times & \times & \\ \text{Equation 2} & \times & \times & & \times \end{array} \tag{2.182}$$

Because there are no structurally singular subsets of equations, the algorithm again does not perform any differentiations, and indicates that there are two dynamic degrees of freedom. However, the canonical form of the system

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_t + \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} \tag{2.183}$$

consists of a differential subsystem of dimension one. Therefore, only one dynamic degree of freedom exists for this system.

The problem lies in the fact that, even though both $\dot{u}_1$ and $\dot{u}_2$ appear in each of the two equations, it is impossible to solve them uniquely for $\dot{u}_1$ and $\dot{u}_2$ in terms of u_1

[8] This is not always the case; it is possible for the index to equal the number of differentiations even when the derivatives of some variables do not appear in the final system.

and u_2. Elimination of $\dot{u}_1$ from an equation also eliminates $\dot{u}_2$, and vice versa. This singularity is numerical, because precisely the same combination of the two variables of interest ($\dot{u}_1$ and $\dot{u}_2$) appears in both equations.

The algorithm may thus also fail to correctly determine the differentiation index in the presence of such numerical singularities. In the chemical engineering community, the number of differentiations returned by the algorithm was for a time assumed to be a lower bound on the true differentiation index. However, it was not widely appreciated that, in the presence of numerical singularities, the algorithm may also perform a *greater* number of differentiations than the true differentiation index, so in fact the algorithm does not provide a bound on the index [71].

As an example where Pantelides' algorithm *overestimates* the differentiation index, consider the following simple system.

$$\begin{aligned} \dot{u}_2 + \dot{u}_3 + u_1 &= 0 \\ \dot{u}_2 + \dot{u}_3 + u_2 &= 0 \\ \dot{u}_4 + \dot{u}_5 + u_3 &= 0 \\ \dot{u}_4 + \dot{u}_5 + u_4 &= 0 \\ u_5 &= 0 \end{aligned} \tag{2.184}$$

Pantelides' algorithm differentiates the final equation in its first iteration. It then differentiates the final three equations on its second iteration. At this point, the algorithm terminates, and no algebraic equations or variables are present, so the expected index is two. However, one differentiation of the last four equations immediately gives $\dot{\mathbf{u}}$ as a continuous function of $\mathbf{u}$ and t, so the true differentiation index is only one.

Despite the fact that, in the presence of numerical singularities, Pantelides' algorithm may not return the true differentiation index, the fact that it is capable of analyzing nonlinear systems and may be applied efficiently to large DAEs has led to its continued use in dynamic process simulators. For chemical engineering models, such numerical singularities appear to be uncommon, although examples have been reported.

2.4 Partial Differential Equations

ODE and DAE systems relate unknown functions of a single variable. If the unknowns are instead functions of more than one variable, the equations are called partial differential equations. Just as differential equations introduce a new set of issues that do not occur with strictly algebraic systems, partial differential equations give rise to rich geometric analyses and to new issues not encountered in differential-algebraic or ordinary differential equations.

2.4.1 Notation and classification

Partial differential equations, or **PDEs**, relate functions of more than one independent variable. Consider a single dependent variable u that is a function of two independent variables t and x. The **partial derivative** of u with respect to x is most often written in one of the following two ways, both of which are equivalent.

$$\frac{\partial u}{\partial x} = u_x \tag{2.185}$$

The notation on the right is more compact, and will be used wherever possible, so for example u_{xx} is the second partial derivative of u with respect to x. u_{tx} is also a second partial derivative, called a **mixed partial derivative**.

$\mathbf{u}_x$ denotes a vector containing the partial derivatives of the elements of $\mathbf{u}$ with respect to x.

$$\mathbf{u}_x = \begin{bmatrix} u_{1_x} \\ u_{2_x} \\ \vdots \\ u_{n_x} \end{bmatrix} \tag{2.186}$$

Similarly $\mathbf{A}_t$ is a matrix formed by differentiating each element of $\mathbf{A}$ once with respect to t.

When the partial derivative of $\mathbf{u}$ with respect to x is to be expressed as a differential operator acting on $\mathbf{u}$, it will be written as

$$\frac{\partial}{\partial x}\mathbf{u} \tag{2.187}$$

The partial differential operator is a linear operator, so for a constant parameter a, two dependent variables v and w, and two independent variables x and y, the following properties hold.

$$\begin{aligned} \frac{\partial}{\partial x}(v+w) &= \frac{\partial}{\partial x}v + \frac{\partial}{\partial x}w \\ \frac{\partial}{\partial x}(av) &= a\frac{\partial}{\partial x}v \\ \frac{\partial}{\partial x}\frac{\partial}{\partial y}v &= \frac{\partial}{\partial y}\frac{\partial}{\partial x}v \end{aligned} \tag{2.188}$$

Consider a general second order partial differential equation in the dependent variable u and two independent variables x and y.

$$f(u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, x, y) = 0 \tag{2.189}$$

If the highest order partial derivatives occur linearly, so that the equation can be rewritten in the form

$$\begin{aligned} a(u, u_x, u_y, x, y)u_{xx} + 2b(u, u_x, u_y, x, y)u_{xy} +& \\ c(u, u_x, u_y, x, y)u_{yy} = d(u, u_x, u_y, x, y)& \end{aligned} \tag{2.190}$$

then it is called **quasilinear**. If a, b, and c depend only on the independent variables, so that the equation may be written as

$$a(x, y)u_{xx} + 2b(x, y)u_{xy} + c(x, y)u_{yy} = h(u, u_x, u_y, x, y) \tag{2.191}$$

it is called **semilinear**. If a, b, and c are constants, the equation is called **linear**. Analogous classifications apply for a first order equation, where a and c are the coefficients of u_x and u_y.

The equation is classified as one of three types based on a **discriminant**, which is $b^2 - ac$.

$$\begin{aligned} b^2 - ac &> 0 \quad \text{hyperbolic} \\ b^2 - ac &= 0 \quad \text{parabolic} \\ b^2 - ac &< 0 \quad \text{elliptic} \end{aligned}$$

A higher order system may always be expressed as a larger first order system, through the introduction of new variables for the higher order terms [40].

Several specific partial differential operators that allow balance equations to be expressed in a very compact form are are often used in fluid dynamics literature [4]. The **gradient** operator ∇ raises the dimensionality of its operand, taking scalars to vectors and vectors to matrices. For the scalar p on a two-dimensional domain,

$$\nabla p = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \tag{2.192}$$

while for the vector[9] $\mathbf{v}$,

$$\nabla \mathbf{v} = \begin{bmatrix} v_{1_x} & v_{1_y} \\ v_{2_x} & v_{2_y} \end{bmatrix} \tag{2.193}$$

Just as the gradient increases the order of its argument by one (taking scalars to vectors and vectors to matrices), the **divergence** operator $\nabla\cdot$ decreases the dimensionality of its argument. Again considering a vector $\mathbf{v}$,

$$\nabla \cdot \mathbf{v} = v_{1_x} + v_{2_y} \tag{2.194}$$

Similarly,

$$\nabla \cdot \mathbf{A} = \begin{bmatrix} A_{11_x} + A_{12_y} \\ A_{21_x} + A_{22_y} \end{bmatrix} \tag{2.195}$$

The **Laplacian** operator ∇^2 is a composition of the gradient and divergence operators. It does not alter the dimensionality of its argument. For the scalar p, recall (2.192) and (2.194); thus by expanding the Laplacian,

$$\nabla^2 p = \nabla \cdot (\nabla p) \tag{2.196}$$

$$= p_{xx} + p_{yy} \tag{2.197}$$

[9] Different authors define the gradient in different ways. The notation presented here [4] is typically used in the fluid dynamics community, with $(\nabla \mathbf{v})_{ij} = \frac{\partial v_i}{\partial x_j}$. Other authors [7] instead define $(\nabla \mathbf{v})_{ij} = \frac{\partial v_j}{\partial x_i}$. The divergence operator is defined to match the gradient operator, so that the Laplacian operator is universally defined in the manner shown in this section.

For **v**, using (2.193) and (2.195) gives us

$$\begin{aligned}\nabla^2\mathbf{v} &= \nabla\cdot(\nabla\mathbf{v}) && (2.198)\\ &= \begin{bmatrix} v_{1_{xx}} + v_{1_{yy}} \\ v_{2_{xx}} + v_{2_{yy}} \end{bmatrix} && (2.199)\end{aligned}$$

The case of a matrix is similar.

The notion of a **directional derivative** is related to coordinate changes. Consider a partial differential equation over the independent variables x and y. To rewrite the equation in terms of new independent variables ξ and η, the chain rule may be used to convert partial differential operators in x and y to equivalent operators in ξ and η.

$$\begin{aligned}\xi &= \xi(x,y) && (2.200)\\ \eta &= \eta(x,y) && (2.201)\end{aligned}$$

so

$$\frac{\partial}{\partial\xi} = \frac{\partial x}{\partial\xi}\frac{\partial}{\partial x} + \frac{\partial y}{\partial\xi}\frac{\partial}{\partial y} \qquad (2.202)$$

This gives the directional derivative along ξ, which is a partial derivative in the new coordinate system, in terms of the derivatives in the x and y directions in the old coordinate system. The first term on the righthand side is the x component of the derivative in the ξ direction, and the second term is the y component.

Directional derivatives are related to **interior** and **exterior derivatives**. Consider a surface defined at a point by its normal vector. The $n-1$ dimensional tangent hyperplane to the surface at that point will be spanned by $n-1$ linearly independent vectors in n-space, called the basis vectors for that surface. Each of these basis vectors is by definition orthogonal to the normal to the surface. The case of a surface in 3-dimensional space, with normal vector **p**, appears in figure 2-3. Derivatives taken in the direction of the basis vectors for the surface are the interior derivatives on that surface. Differentiation along the normal gives the exterior derivative.

Given a vector of m independent variables $\mathbf{x}\in\mathbb{R}^m$ and a vector of n dependent variables $\mathbf{u}(\mathbf{x}):\mathbb{R}^m\rightarrow\mathbb{R}^n$, the **Jacobian** of **u** with respect to **x**, written $\mathbf{J}(\mathbf{u},\mathbf{x})$, is

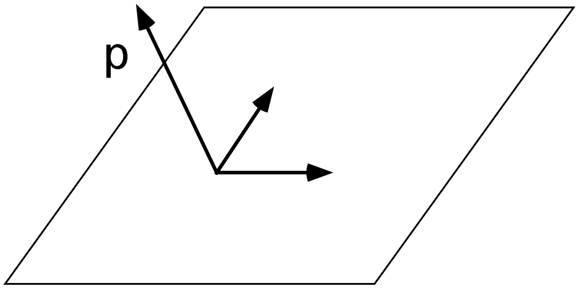

Figure 2-3: Normal and basis vectors for a plane

a matrix containing the partial derivatives of each element of **u** with respect to each independent variable x_j.

$$J(\mathbf{u}, \mathbf{x})_{i,j} = \frac{\partial u_i}{\partial x_j} \tag{2.203}$$

or

$$\mathbf{J}(\mathbf{u}, \mathbf{x}) = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \cdots & \frac{\partial u_1}{\partial x_m} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \cdots & \frac{\partial u_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_n}{\partial x_1} & \frac{\partial u_n}{\partial x_2} & \cdots & \frac{\partial u_n}{\partial x_m} \end{bmatrix} \tag{2.204}$$

2.4.2 Superposition and linear systems

Consider a first order homogeneous linear partial differential equation.

$$a u_t + b u_x = 0 \tag{2.205}$$

Suppose that there are two functions v and w that satisfy this equation. Then, because partial differentiation is a linear operator, any linear combination of v and w also satisfies the equation.

$$a(v + w)_t + b(v + w)_x = 0 \tag{2.206}$$

Combining two or more different solutions to form another solution is called **superposition**. It applies to inhomogeneous equations as well. If there are two or more solutions to the corresponding homogeneous equation, a superposition of the particular solution and the homogeneous solutions will also satisfy the equation.

2.4.3 Separation of Variables

A common solution technique for linear partial differential equations is **separation of variables**. An assumption is made about the form of the solution; typically, that it is the product of a function of t only with a function of x only.

$$u(x,t) = f(t)g(x) \tag{2.207}$$

This expression is then substituted into the partial differential equation, and terms are rearranged so that a function of x only appears on one side of the equation and a function of t only appears on the other. As an example, consider the **heat equation**

$$u_t - u_{xx} = 0 \tag{2.208}$$

and substitute in the expression above (2.207), which yields

$$\begin{aligned} f'(t)g(x) - f(t)g''(x) &= 0 \\ \Rightarrow \quad \frac{f'(t)}{f(t)} &= \frac{g''(x)}{g(x)} = -\lambda \end{aligned} \tag{2.209}$$

for a constant λ. This is because a function of t can equal a function of x for all values of independent variables t and x only if both functions are constants.

The functions $f(t)$ and $g(x)$ are the solutions of the following two ordinary differential equations.

$$\begin{aligned} \frac{g''(x)}{g(x)} &= -\lambda \\ g''(x) &= -\lambda g(x) \\ g(x) &= a\cos(\sqrt{\lambda}x) + b\sin(\sqrt{\lambda}x) \end{aligned} \tag{2.210}$$

$$\begin{aligned} \frac{f'(t)}{f(t)} &= -\lambda \\ f'(t) &= -\lambda f(t) \\ f(t) &= e^{-\lambda t} \end{aligned} \tag{2.211}$$

A unique solution of this form (2.207) is of course determined by initial and boundary conditions. For example, consider the domain $0 \le x \le \pi$ and $0 \le t$, and boundary conditions $u(0,t) = u(\pi,t) = 0$ and $u(x,0) = f(x)$. In order for the solution to always satisfy the boundary conditions, $a = 0$ and $\lambda = n^2$ in the expression for g (2.210), where n is any integer.

At this point b is still undetermined, but $u(t,x)$ will be a superposition of functions of the form

$$u_n(x,t) = b_n e^{-n^2 t} \sin(nx) \tag{2.212}$$

If these functions are evaluated at $t = 0$, the superposition must equal the initial condition, so[10]

$$\sum_{n=-\infty}^{\infty} b_n \sin(nx) = f(x) \tag{2.213}$$

Assuming that $f(x)$ can be represented by an infinite sine series, this will determine unique values for all b_n [18].

2.4.4 Solution via Fourier transform

The **Fourier transform in** x of a function $u(t,x)$, denoted by $\hat{u}(t,\omega)$, is defined as follows.

$$\hat{u}(t,\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} u(t,x) e^{-i\omega x} dx \tag{2.214}$$

The Fourier transform is a function of the wavenumber (or frequency) ω. If $u(t,x)$ is considered to be a superposition of waves, $\hat{u}(t,\omega)$ gives the amplitude of the wave with wavenumber ω.

The **inverse Fourier transform in x** is

$$u(t,x) = \int_{-\infty}^{\infty} \hat{u}(t,\omega) e^{i\omega x} d\omega \tag{2.215}$$

If this expression is substituted into a partial differential equation, partial derivatives with respect to x are given by multiples of $i\omega$. For example, substitution of this

[10] Because $\sin(-x) = -\sin(x)$, the coefficients b_n for the $\sin(-x)$ and $\sin(x)$ terms are sometimes combined, and only positive integers n are considered.

expression into the heat equation gives

$$\begin{aligned}\int_{-\infty}^{\infty} \hat{u}_t(t,\omega)e^{i\omega x}d\omega - \int_{-\infty}^{\infty} -\omega^2\hat{u}(t,\omega)e^{i\omega x}d\omega = 0\\ \int_{-\infty}^{\infty} \left(\hat{u}_t(t,\omega) + \omega^2\hat{u}(t,\omega)\right) e^{i\omega x}d\omega = 0\end{aligned} \tag{2.216}$$

so

$$\begin{aligned}\hat{u}_t(t,\omega) + \omega^2\hat{u}(t,\omega) = 0\\ \hat{u}(t,\omega) = \hat{u}(0,\omega)e^{-\omega^2 t}\end{aligned} \tag{2.217}$$

One could then use the inverse Fourier transform to obtain the solution in (t, x) space. However, the solution in **Fourier space** (t, ω), (also called the **frequency domain**), provides useful information. Here, the solution in Fourier space shows that, for an initial condition that is a superposition of waves, the highest frequency components of the superposition decay the fastest in t. Every component decays, at a rate proportional to the square of the frequency.

The L_2 norm of a function is equal to that of its Fourier transform.

$$||\mathbf{f}(t,\cdot)|| = ||\hat{\mathbf{f}}(t,\cdot)|| \tag{2.218}$$

This is known as **Parseval's equation**.

2.4.5 Linear stability analysis

The concept behind the term **linear stability** starts with the idea of a **perturbation**. A perturbation is a small change. If ϵ is a perturbation of a, then ϵ is assumed to be small when compared to a, so that $a + \epsilon \approx a$.

For an ordinary differential equation, an initial condition is a scalar constant u_0. Stability analysis of the solution of an ordinary differential equation asks the question "what happens to the solution if the initial condition is perturbed?"

The solution to an ordinary differential equation is uniquely determined by an initial condition. Perturbing that initial condition produces a different solution. The original solution is said to be **stable** if the difference between the solution determined by the perturbed initial condition and the original solution is never greater than a linear function of the size of the perturbation. This is also called **stability in the sense of Lyapunov**; another type of stability employed frequently in process control applications is the stronger notion of **asymptotic stability**, which requires that the difference between any perturbed solution and the original solution must decay to zero.

More precisely, let $u(t)$ be the solution determined by an initial condition u_0, and let $u^*(t)$ be the solution determined by the initial condition $u_0+\epsilon$. Then $u(t)$ is stable if there exists some constant k such that

$$|u(t) - u^*(t)| \leq k|\epsilon| \tag{2.219}$$

for all $t \geq t_0$.

For example, if $u(t) = u_0 e^{ct}$ and $u^*(t) = (u_0 + \epsilon)e^{ct}$, then

$$|u(t) - u^*(t)| = |\epsilon e^{ct}| = |\epsilon|\ |e^{ct}| \tag{2.220}$$

Now, if $c \leq 0$, then for all $t \geq 0$, $|e^{ct}| \leq 1$, so

$$|u(t) - u^*(t)| \leq |\epsilon| \tag{2.221}$$

and the stability condition (2.219) is met for any $k \geq 1$. However, if $c > 0$, there is no constant k for which $|e^{ct}| \leq k$ for all $t \geq 0$. The solution is therefore stable iff $c \leq 0$.

For a partial differential equation in t and x, an initial condition is now a function of x rather than simply a scalar; $u(0, x) = f(x)$. A perturbation in this initial condition is also a scalar function of x. One can ask the same question about stability, "what happens to the solution $u(t, x)$ if the initial condition $f(x)$ is perturbed?"

One could perform a stability analysis that is similar to that for an ordinary differential equation, and look at how the solution depends on the size of a perturbation $g(x)$ in the initial condition $f(x)$. A measure of the size of a function is its L_2 norm. If $u(t, x)$ is the solution determined by the initial condition $u(0, x) = f(x)$, and $u^*(t, x)$ is the solution determined by $u(0, x) = f(x) + g(x)$, then $u(t, x)$ is considered to be stable if there exists some constant k such that

$$||u(t, \cdot) - u^*(t, \cdot)|| \leq k||g(\cdot)|| \tag{2.222}$$

for all $t \geq 0$.

2.4.6 Well posed initial-boundary value problems

Let a **problem** be defined as a set of partial differential equations together with the specification of a domain and initial and boundary conditions. Initial and boundary conditions are considered to be **data** if they are used to fix the values of constants in a solution; they are not considered to be data if they are used to select a subset of functions from which a superposition is constructed.

For example, in the heat equation example above (2.208 - 2.212), if the solution is to be built as a superposition of sines and cosines, the boundary conditions are used to eliminate all cosines, and all sines for which the domain length is not an integer multiple of its half-period, and so the boundary conditions are not considered to be data. The initial condition is used to fix the values of arbitrary constants, and so is considered to be data. A different solution method for the same problem might use the initial and boundary conditions in a different manner, so the classification of some initial and boundary conditions as data is specific to each problem and solution method.

A problem is said to be **well-posed** if a solution exists, that solution is unique, and the solution depends continuously on its data. Existence and uniqueness will be considered later, and their meaning is intuitively clear. Continuous dependence on data has no analogue in the study of differential equations, and will be examined in more detail in the next section.

2.4.7 Continuous dependence on data

The unspoken rule of a perturbation ϵ to another quantity a is that ϵ and a are of the same type. If $a \in \mathbb{R}$, then $\epsilon \in \mathbb{R}$. A perturbation $\mathbf{p}$ of a vector $\mathbf{v}$ is itself a vector of the same size; $\mathbf{v} \in \mathbb{R}^n \Rightarrow \mathbf{p} \in \mathbb{R}^n$. A function $f(t)$ may be perturbed by another function $g(t)$.

Because the initial condition is now a function, rather than a scalar, there is more to a perturbation of the initial condition than simply its magnitude. Consider, for example, two perturbations $g_1(x) = \sin(x)$ and $g_2(x) = \sin(2x)$, and an infinite domain in x. Clearly

$$||g_1(\cdot)|| = ||g_2(\cdot)|| \tag{2.223}$$

In fact, the magnitudes of any two sine waves are identical, regardless of frequency.

Linear stability analysis examines the dependence of the solution on the *magnitude* of a perturbation in the initial condition. Analysis of **continuous dependence on data** looks instead at the dependence of the solution on the *frequency* of a perturbation in the initial condition. If the change in a solution can be bounded independently of the frequency of a perturbation, it is said to depend continuously on its data. It does not need to be stable in order to depend continuously on its data. The reverse is not true, however; if it is not possible to bound the change in the solution independently of the frequency of a small perturbation to the initial data, then the solution is unstable for at least some perturbations of arbitrarily small magnitude.

An evolution problem (in t) depends continuously on its initial data if small changes in that data produce bounded (but not necessarily small!) changes in the solution at later times. If

$$||\mathbf{u}(t,\cdot)|| \leq C_t||\mathbf{u}(0,\cdot)|| \tag{2.224}$$

holds for all $\mathbf{u}(0,x)$ in some norm, such as the L^2 norm, and for some function[11] C_t that is independent of the solution but may depend on t, then the solution depends continuously on its data [81].

If proper initial and boundary conditions are provided for the system, but the dependence of the solution on the initial data only satisfies an estimate of the form

$$||\mathbf{u}(t,\cdot)|| \leq C_t||\mathbf{u}(0,\cdot)||_{H^q} \tag{2.225}$$

where the H^q norm is the L^2 norm of a function and its derivatives in x of order q or lower, given by

$$||\mathbf{f}||^2_{H^q} = \sum_{0\leq\nu\leq q} || \left(\frac{\partial}{\partial x}\right)^{\nu} \mathbf{f}||^2 \tag{2.226}$$

[11] The literature refers to this function of t as a "contant that may depend on t" [81]. The notation chosen here is consistent with the literature; C_t is a function $f(t)$, not a constant or a partial derivative with respect to t.

then the system is said to be **weakly well-posed**. Because the solution to a weakly well-posed system depends on derivatives of the initial conditions, higher order methods are sometimes recommended for weakly well-posed problems [44]. Briefly, this is because a given finite difference or finite element mesh can resolve a finite maximum frequency perturbation, and this frequency increases as the mesh is refined. The discretization must force the error to zero faster than the increasing frequency perturbations distort the solution. Note that a weakly well-posed system is a special type of ill-posed, rather than well-posed, system. An ill-posed system that is not weakly well-posed may be referred to as **strongly ill-posed**.

The primary tool for examining continuous dependence on data is the Fourier transform, together with Parseval's equation. Consider, for example, the heat equation.

$$u_t - cu_{xx} = 0 \tag{2.227}$$

The solution $u(t,x)$ depends continuously on its initial data iff there is a bounded C_t that is independent of the solution, that bounds $u(t,x)$ in terms of $u(0,x)$ as above (2.224).

The Fourier transform produces

$$\hat{u}_t + c\omega^2\hat{u} = 0 \tag{2.228}$$

for which the solution is

$$\hat{u}(t,\omega) = \hat{u}(0,\omega)e^{-c\omega^2 t} \tag{2.229}$$

Taking the norm of both sides,

$$||\hat{u}(t,\cdot)|| = ||\hat{u}(0,\cdot)e^{-c\omega^2 t}|| \leq |e^{-c\omega^2 t}|\; ||\hat{u}(0,\cdot)|| \tag{2.230}$$

If $c \geq 0$, note that $e^{-c\omega^2 t} \leq 1$ for all $\omega \in \mathbb{R}$ and all $t > 0$, so if $C_t = 1$,

$$|e^{-c\omega^2 t}|\; ||\hat{u}(0,\cdot)|| \leq C_t||\hat{u}(0,\cdot)|| \tag{2.231}$$

Substituting this result back into the original inequality,

$$||\hat{u}(t,\cdot)|| \leq C_t||\hat{u}(0,\cdot)|| \tag{2.232}$$

and by Parseval's equation

$$||u(t,\cdot)|| \leq C_t ||u(0,\cdot)|| \tag{2.233}$$

so if $c \geq 0$, the solution depends continuously on its data.

This approach may also be used to analyze the dependence on data of the solution to more general *systems* of partial differential equations. First, the Fourier transform is used to obtain a system of the form

$$\hat{\mathbf{u}}_t(t,\omega) + \mathbf{P}(t,\omega)\hat{\mathbf{u}}(t,\omega) = \mathbf{0} \tag{2.234}$$

for which the solution is

$$\hat{\mathbf{u}}(t,\omega) = e^{-\mathbf{P}(t,\omega)t}\hat{\mathbf{u}}(0,\omega) \tag{2.235}$$

If there exists some function of t, again written C_t and given by $C_t = Ke^{\alpha t}$ with constants K and α, for which

$$||e^{\mathbf{P}(t,\omega)t}|| \leq C_t = Ke^{\alpha t} \tag{2.236}$$

holds for all possible values of ω and for $t \geq 0$, then Parseval's equation may be used as above to show that the solution depends continuously on its initial data. If no such C_t can be found, but there exist constants K, α and positive constant q such that

$$||e^{\mathbf{P}(t,\omega)t}|| \leq Ke^{\alpha t}(1+\omega^q) \tag{2.237}$$

holds for all ω and for $t \geq 0$, the system does not depend continuously on its data and is instead weakly well-posed [44].

As an example of this analysis, consider a system of the form

$$\mathbf{u}_t + \mathbf{B}\mathbf{u}_x = \mathbf{0} \tag{2.238}$$

Taking the Fourier transform produces

$$\hat{\mathbf{u}}_t + i\omega\mathbf{B}\hat{\mathbf{u}} = \mathbf{0} \tag{2.239}$$

No further manipulation is required to produce a system of the form under consideration (2.234), with

$$\mathbf{P}(t, \omega) = i\omega\mathbf{B} \tag{2.240}$$

The original system (2.238) is called **hyperbolic** iff all eigenvalues of $\mathbf{B}$ are strictly real and distinct. If this is true, there exists a constant matrix $\mathbf{S}$ such that $\mathbf{B} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$. Then

$$\begin{aligned}||e^{-i\omega\mathbf{B}t}|| &= ||\mathbf{S}e^{-i\omega\mathbf{\Lambda}t}\mathbf{S}^{-1}|| \\ &\leq ||\mathbf{S}||\;||e^{-i\omega\mathbf{\Lambda}t}||\;||\mathbf{S}^{-1}|| \\ &\leq k||e^{-i\omega\mathbf{\Lambda}t}||\end{aligned} \tag{2.241}$$

Because all eigenvalues of $\mathbf{B}$ are strictly real, all elements of the diagonal matrix $-i\omega\mathbf{\Lambda}t$ are purely imaginary, and

$$e^{-i\omega\mathbf{\Lambda}t} = \begin{bmatrix} e^{-i\omega\lambda_1 t} & & \\ & \ddots & \\ & & e^{-i\omega\lambda_n t} \end{bmatrix} \tag{2.242}$$

Because the norm of a matrix is bounded from above by the magnitude of its largest element (2.54), and the magnitude of the exponential of any pure imaginary number (2.102) is always one,

$$||e^{-i\omega\mathbf{\Lambda}t}|| \leq n \tag{2.243}$$

for all values of t or ω. Therefore, the system depends continuously on its initial data, because for $K = kn$ and $\alpha = 0$,

$$||e^{-i\omega\mathbf{B}t}|| \leq kn \leq Ke^{\alpha t} = C_t \tag{2.244}$$

for all ω.

Now, if all eigenvalues of $\mathbf{B}$ are strictly real, but one or more has geometric multiplicity greater than unity, then the system does not depend continuously on its data and is weakly well-posed. If any eigenvalue of $\mathbf{B}$ has a nonzero imaginary part, the solution also fails to depend continuously on its data, but is strongly ill-posed.

So in summary, the stability of a solution that depends continuously on its data has a "worst case". That worst case may be very unstable, but there is a "worst" perturbation. A solution that does not depend continuously on its data has no worst case; for every frequency of perturbation that causes the difference from the original solution to grow quickly, there is another one that grows even more quickly.

2.4.8 Semilinear and quasilinear systems

The previous section considered only linear systems. It said nothing about quasilinear or nonlinear systems, such as

$$\mathbf{u}_t + \mathbf{B}(\mathbf{u})\mathbf{u}_x = \mathbf{0} \tag{2.245}$$

Because the system is not linear, its properties may change with different values of $\mathbf{u}$, t, and x. A general approach to analyzing quasilinear and nonlinear problems is to linearize the system at a nominal value of interest $\mathbf{u}_0$, and then examine the properties of the resulting (linear) system in the manner described in the previous section. The original system is then said to depend continuously on its data at $\mathbf{u}_0$ if it can be shown that the problems that are obtained by linearizing at all functions near $\mathbf{u}_0$ depend continuously on their data.

For a quasilinear system, one approach is to simply evaluate $\mathbf{B}(\mathbf{u})$ at the nominal value of interest $\mathbf{u} = \mathbf{u}_0$. For the example system under consideration, this gives

$$\mathbf{u}_t + \mathbf{B}(\mathbf{u}_0)\mathbf{u}_x = \mathbf{0} \tag{2.246}$$

This resulting system is called the **frozen coefficient system**.

A more rigorous approach, that may also be applied to nonlinear systems, is **formal linearization**. Under this approach, each dependent variable is assumed to have the form of a small unknown perturbation to a function with a known value. Substitution then gives the system that governs the behavior of the solution near the nominal (known) value.

The formal linearization of a quasilinear system may differ from the frozen coefficient system, because formal linearization may introduce lower order terms. To illustrate this, consider the following example [44]. Suppose the solution u to Burger's equation

$$u_t - uu_x - \epsilon u_{xx} = 0, \quad \epsilon > 0 \tag{2.247}$$

is the sum of a known smooth function $U(t,x)$ and a small correction $v(t,x)$. Substitution of

$$u(t,x) = U(t,x) + v(t,x) \tag{2.248}$$

into the original equation produces the formal linearization at U

$$v_t - Uv_x - \epsilon v_{xx} - U_x v - vv_x = F \tag{2.249}$$

where F is a known function of t and x, given by

$$F = UU_x + \epsilon U_{xx} - U_t \tag{2.250}$$

Because v is considered to be a small correction to U, the quadratic term vv_x may be dropped from $(U+v)v_x$. The equation that governs small perturbations v about the nominal operating value U is then

$$v_t - Uv_x - \epsilon v_{xx} - U_x v = F \tag{2.251}$$

which is the same as the frozen coefficient system at U, perturbed by the additional linear term $U_x v$. Note that $F = 0$ if U solves the original equation exactly.

If it can be shown that a particular class of linear systems depends continuously on its data in the presence of arbitrary lower-order forcing terms, well-posedness of the frozen coefficient system implies well-posedness of the formal linearization. For example, it has been shown [81] that a system of the form

$$\mathbf{u}_t + \mathbf{B}\mathbf{u}_x + \mathbf{C}\mathbf{u} = \mathbf{f}(t,x) \tag{2.252}$$

depends continuously on its data iff it is hyperbolic, which is true iff all eigenvalues of $\mathbf{B}$ are real and distinct. This means that a system of the form

$$\mathbf{u}_t + \mathbf{B}(\mathbf{u})\mathbf{u}_x + \mathbf{C}\mathbf{u} = \mathbf{f}(t,x) \tag{2.253}$$

depends continuously on its data at $\mathbf{u}_0$ iff the corresponding frozen coefficient system depends continuously on is data. In other words, the dependence of the formal linearization on its data is not sensitive to the lower order terms introduced by the linearization, and so it is sufficient to analyze the eigenvalues of $\mathbf{B}$ evaluated at $\mathbf{u}_0$ in order to determine whether or not the solution to the original system depends continuously on its data at $\mathbf{u}_0$.

Weakly well-posed systems are *not* insensitive to the introduction of lower order terms. Consider, for example, the following simple system [44].

$$\mathbf{u}_t + \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \mathbf{u}_x = \mathbf{0} \tag{2.254}$$

The coefficient matrix has a single strictly real eigenvalue (unity) of geometric multiplicity 2, and so is weakly well-posed. However, upon introduction of a single linear term, the resulting system may be strongly ill-posed. Consider

$$\mathbf{u}_t + \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \mathbf{u}_x + \begin{bmatrix} 0 & \epsilon \\ 0 & 0 \end{bmatrix} \mathbf{u} = \mathbf{0} \tag{2.255}$$

The Fourier transform of the system is

$$\hat{\mathbf{u}}_t + \begin{bmatrix} i\omega & \epsilon \\ i\omega & i\omega \end{bmatrix} \hat{\mathbf{u}} = \mathbf{0} \tag{2.256}$$

The eigenvalues λ of the coefficient matrix are given by

$$\lambda = i\omega \pm \sqrt{i\epsilon\omega} \tag{2.257}$$

which have a nonzero imaginary part. Incorporation of a linear term has produced a strongly ill-posed system from a system that was weakly well-posed.

2.4.9 The characteristic form of a hyperbolic equation

A hyperbolic partial differential equation in two independent variables t and x is equivalent to an ODE along special curves in the t, x plane. The transformation to, and interpretation of, this ODE is called **characteristic analysis**.

Consider the **one-way wave equation** on a finite domain.

$$u_t + cu_x = 0, \quad a \leq x \leq b, \;\; t \geq 0 \tag{2.258}$$

Suppose that the solution to the one-way wave equation is carried forward in time along specific curves in the (t, x) plane. Along these curves, x and t vary with distance along the curve (call this distance s). Because x and t now depend on s, u is now a function of only one variable: $u = u(x(s), t(s))$. Therefore by the chain rule

$$\frac{du}{ds} = u_t \frac{dt}{ds} + u_x \frac{dx}{ds} \tag{2.259}$$

This may be thought of as the directional derivative of u in the s direction. Rearranging terms just slightly gives

$$\frac{du}{ds} = \frac{dt}{ds} u_t + \frac{dx}{ds} u_x \tag{2.260}$$

By inspection, the one-way wave equation (2.258) is equivalent to an ODE in s

$$\frac{du}{ds} = 0 \tag{2.261}$$

where the curve s is defined by

$$\frac{dt}{ds} = 1, \quad \frac{dx}{ds} = c \tag{2.262}$$

Proceeding one final step, one can eliminate s from the ordinary differential equation (2.261) and characteristic curve definition (2.262), and write the one-way wave equation as an ODE along a direction in the (t, x) plane.

$$\frac{du}{dt} = 0 \quad \text{along} \quad dx = c\, dt \tag{2.263}$$

This is the **characteristic form** of the equation.

The original partial differential equation is thus equivalent to an ordinary differential equation when one follows the solution in the s-direction. As such, given an initial condition at some point, one can advance the solution from that point in the direction of s. The solution is simple; integrating the characteristic form (2.263) once gives

$$u(t) = k_1 \quad \text{along} \quad x = ct + k_2 \tag{2.264}$$

with k_1 an arbitrary constant determined by an initial condition on u, and k_2 determined by the location in the (x,t) plane where that initial condition is enforced. This solution propagates unchanged in the s-direction.

For constant c, the s-direction is a straight line in the (x,t) plane. From the characteristic form of the one-way wave equation, clearly

$$\frac{dx}{dt} = c \tag{2.265}$$

This is the **characteristic direction** in the (x,t) plane. The characteristic direction, or more simply the characteristic, of the equation is the direction in which information travels. It gives the path of a signal, which for the one-way wave equation is the value of the dependent variable u.

This interpretation of characteristics as **signal trajectories** gives some insight into the question of determining appropriate boundary conditions for partial differential equations. Suppose $c = 1$. The characteristics are then straight lines, with slope of one. The situation appears graphically in figure 2-4. The value of u given by the initial condition travels along the characteristics, so that if the initial condition is given by

$$u(x,0) = f(x), \quad a \le x \le b \tag{2.266}$$

then

$$u(x,t) = f(x-t), \quad a \le x - t \le b \tag{2.267}$$

In other words, the initial solution travels to the right with a speed of 1. The area in grey is the region in which the solution is determined solely by the initial condition, and is called the **domain of influence** of the initial condition.

Since the solution is carried along the characteristics, the solution at $x = b$ in the grey region is given by

$$u(b,t) = f(b-t), \quad t \le b \tag{2.268}$$

This means that a boundary condition cannot independently set the value of u at b; characteristics already carry enough information to b from the interior of the domain to fully determine the solution there.

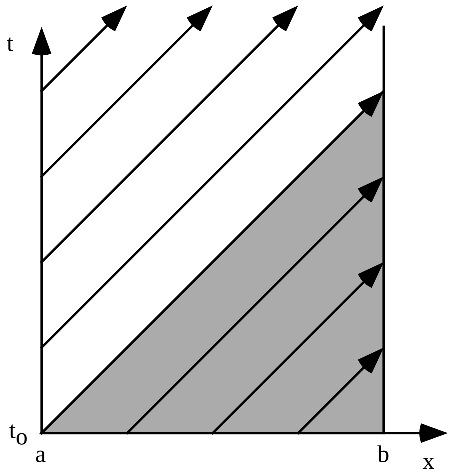

Figure 2-4: Plot of characteristics for one-way wave equation

The case is different at $x = a$. Here, the solution is not determined by the initial condition, because the characteristics carry information away from the boundary and into the domain. A boundary condition is thus required at a in order to fully determine the solution over the whole domain for all time. This boundary condition will set the solution in the white region of figure 2-4. Calling the boundary condition g,

$$u(a,t) = g(t), \quad t > 0 \tag{2.269}$$

one can trace the solution at any point in the white region back along a characteristic to a. This gives

$$u(x,t) = g(t-(x-a)) \quad x - t < a \tag{2.270}$$

This requirement, that a boundary condition must be specified wherever a characteristic enters the domain from the boundary, is a general result [18]. For systems of equations, where there are families of characteristics at every point, one boundary condition is required for every characteristic directed from the boundary into the domain, as will be seen in the next section.

2.4.10 The characteristic form of a hyperbolic system

The signal trajectory interpretation of characteristics becomes more interesting when applied to systems of equations. Consider a linear first order system of n partial

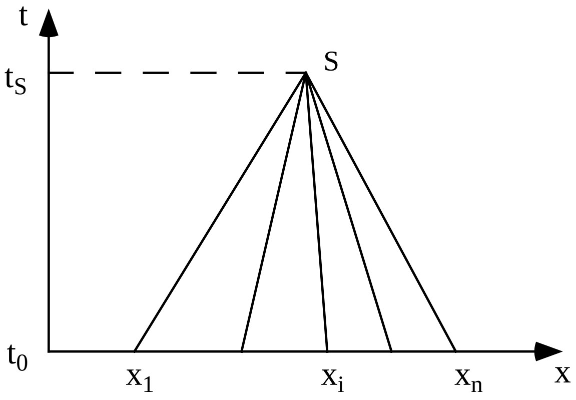

Figure 2-5: Solution at a point determined by characteristics

differential equations of the following form.

$$\mathbf{u}_t + \mathbf{B}\mathbf{u}_x = \mathbf{0} \tag{2.271}$$

The system is assumed to be hyperbolic, which means that $\mathbf{B}$ has a complete set of left eigenvectors $\mathbf{l}_i$ and that all eigenvalues λ_i are strictly real.

If the system is multiplied on the left by a left eigenvector $\mathbf{l}_i$, making the substitution $\mathbf{l}_i\mathbf{B} = \lambda_i\mathbf{l}_i$ produces

$$\mathbf{l}_i(\mathbf{u}_t + \lambda_i\mathbf{u}_x) = \mathbf{0} \tag{2.272}$$

Now let $v_i = \mathbf{l}_i\mathbf{u}$, which gives

$$v_{i_t} + \lambda_i v_{i_x} = 0 \tag{2.273}$$

This is a one-way wave, which is equivalent to

$$\frac{dv_i}{dt} = 0 \quad \text{along} \quad dx = \lambda_i dt \tag{2.274}$$

Performing the same steps for each left eigenvector produces a set of n ODEs along n directions in the (t, x) plane. Taken together, these ODEs are the characteristic form of the original hyperbolic system.

The solution at point S is determined by the information carried to it along the characteristics, as shown in figure 2-5. The solution for a single wave is simply

$$v_i(x_S, t_S) = v_i(x_i, t_0) \tag{2.275}$$

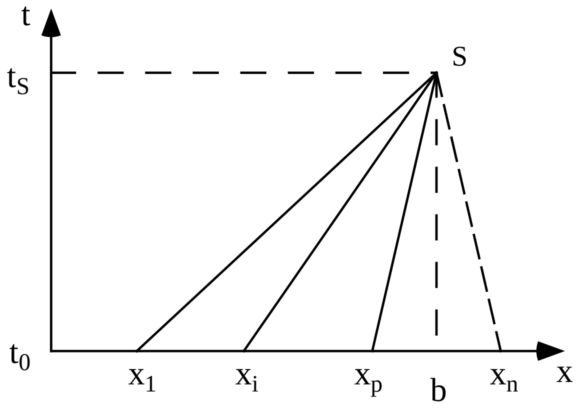

Figure 2-6: Solution at a point partially determined by characteristics

Let $\tilde{\mathbf{v}}$ be a vector that consists of the values of $\mathbf{v}$ at the **feet of the characteristics**; in other words, $\tilde{v}_i = v_i(x_i, t_0)$. Then $\mathbf{v}(x_S, t_S)$ is given by

$$\mathbf{v}(x_S, t_S) = \tilde{\mathbf{v}} \tag{2.276}$$

This may be written in terms of the original variables $\mathbf{u}$. Let $\mathbf{L}$ be the matrix of left eigenvectors of $\mathbf{B}$, so that $\mathbf{v} = \mathbf{L}\mathbf{u}$. Then

$$\begin{aligned} \mathbf{L}\mathbf{u} &= \tilde{\mathbf{v}} \\ \mathbf{u}(x_S, t_S) &= \mathbf{L}^{-1}\tilde{\mathbf{v}} \end{aligned} \tag{2.277}$$

Now, what if S is a point on a domain boundary? For example, consider the situation shown in figure 2-6. It is in general not possible to determine the value of $\mathbf{u}(x_n, t_0)$ outside the domain, so only the characteristics with non-negative slope in the (x, t) plane carry a known signal to S.

The characteristics with non-negative slope partially determine the values of the dependent variables at S, however, and this places restrictions on possible boundary conditions that may be enforced there. If, in figure 2-6, there are p non-negative characteristics, then the rows of (2.277) that correspond to those characteristics will partially determine $\mathbf{u}(b, t_S)$.

$$\begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \vdots \\ \mathbf{l}_p \end{bmatrix} \mathbf{u}(x_S, t_S) = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \vdots \\ \tilde{v}_p \end{bmatrix} = \begin{bmatrix} \mathbf{l}_1\mathbf{u}(x_1, t_0) \\ \mathbf{l}_2\mathbf{u}(x_2, t_0) \\ \vdots \\ \mathbf{l}_p\mathbf{u}(x_p, t_0) \end{bmatrix} \tag{2.278}$$

Rewritten using more compact notation, the equations become

$$\mathbf{C}\mathbf{u}_S = \mathbf{g} \tag{2.279}$$

The coefficients of this system have been assembled into the $p \times n$ element matrix $\mathbf{C}$, and $\mathbf{g}$ is a p element vector constructed from the values of $\mathbf{u}$ at the feet of the non-inward directed characteristics.

If $p < n$, the system (2.279) is underdetermined. In order to uniquely determine $\mathbf{u}(x_S, t_S)$, $n - p$ boundary conditions must be specified. These boundary conditions must be independent of the information carried to $\mathbf{S}$ along characteristics (2.279) and of each other. If the boundary conditions have the form

$$\mathbf{G}\mathbf{u}_S = \mathbf{h} \tag{2.280}$$

then they determine a unique solution iff

$$\begin{vmatrix} \mathbf{C} \\ \mathbf{G} \end{vmatrix} \neq 0 \tag{2.281}$$

2.4.11 Characteristics as discontinuity traces

In addition to their interpretation as signal trajectories, characteristics may also be viewed as **discontinuity traces**. A discontinuity in the solution, whether in the value of a dependent variable or one of its derivatives, may only propagate with special velocities across the domain. One can think of a discontinuity as a special signal, that travels along characteristics but that has special mathematical properties that assist calculation. This interpretation of characteristics is important for analysis in multiple spatial dimensions.

A two dimensional dynamic system such as (2.271) is one in which the independent variables are time t and one other variable x, called the spatial variable. The values of the dependent variables are distributed over the domain in x and change with time, governed by a system of partial differential equations. Typically, initial conditions are set for the equations by specifying $\mathbf{u}$ along x at some time t_o. The equations are then solved for $\mathbf{u}_t$ using the known values of $\mathbf{u}$ and $\mathbf{u}_x$, and integrated to advance the solution in time. If, however, the equations cannot be solved, then $\mathbf{u}_t$ is undefined at $x = x_o$, and there may be a discontinuity in $\mathbf{u}$ across the line $x = x_o$ in the (x, t) plane.

The generalization of the concept of initial conditions from domains in one independent variable to multiple independent variables is **Cauchy data**. Rather than giving the value of $\mathbf{u}$ along the line $x = x_o$, $\mathbf{u}$ may be specified on some arbitrary curve in the (x, t) plane. The derivative of $\mathbf{u}$ is also known in the direction of the curve, and the equations must be used to determine the value of the derivative across the curve. This result is integrated to advance the solution away from the curve [48].

For example, suppose one has a system of two partial differential equations in two unknowns, u and v, as shown below.

$$\begin{aligned} u_t + b_{11}u_x + b_{12}v_x &= h_1 \\ v_t + b_{21}u_x + b_{22}v_x &= h_2 \end{aligned} \tag{2.282}$$

Now, define a curve L in the (x, t) plane by

$$x = L(t) \tag{2.283}$$

$$\frac{dx}{dt} = L' = \lambda(t) \tag{2.284}$$

and suppose that Cauchy data for u and v is given on this curve, so that the first total differentials of u and v are known along L. Now $u = u(x(t), t)$ and $v = v(x(t), t)$, so by the definition of the total differential,

$$\begin{aligned} \frac{du}{dt} &= u_t + \lambda u_x && (2.285) \\ \frac{dv}{dt} &= v_t + \lambda v_x && (2.286) \end{aligned}$$

Using these equations to eliminate u_t and v_t from the system gives conditions on λ under which it is impossible to determine the partial derivatives with respect to x. If those conditions are met, a discontinuity may exist across L.

Solving (2.285) and (2.286) for partial derivatives with respect to t

$$\begin{aligned} u_t &= \frac{du}{dt} - \lambda u_x && (2.287) \\ v_t &= \frac{dv}{dt} - \lambda v_x && (2.288) \end{aligned}$$

and substituting the result into the original PDE system (2.282) yields a system of two equations in two unknowns u_x and v_x

$$\begin{aligned} (b_{11} - \lambda)u_x + b_{12}v_x &= h_1 - \frac{du}{dt} && (2.289) \\ b_{21}u_x + (b_{22} - \lambda)v_x &= h_2 - \frac{dv}{dt} && (2.290) \end{aligned}$$

that does not uniquely determine u_x and v_x if and only if

$$\begin{vmatrix} b_{11} - \lambda & b_{12} \\ b_{21} & b_{22} - \lambda \end{vmatrix} = 0 \qquad (2.291)$$

This equation is called the **characteristic condition**. The directions λ that are characteristic are the solutions to this equation. For equations in more than two dependent variables, of the form

$$\mathbf{u}_t + \mathbf{B}\mathbf{u}_x = \mathbf{f} \qquad (2.292)$$

the characteristics λ are the eigenvalues of $\mathbf{B}$.

$$\left|\mathbf{B} - \lambda\mathbf{I}\right| = 0 \qquad (2.293)$$

Had L been defined instead as

$$t = L(x) \qquad (2.294)$$

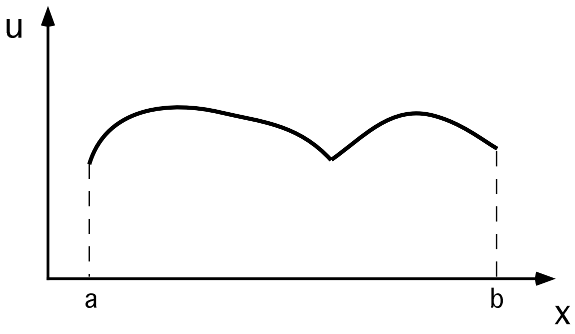

Figure 2-7: C^0 discontinuous solution in one dimension

the characteristic condition would have been

$$\left| \mathbf{I} - \lambda \mathbf{B} \right| = 0 \tag{2.295}$$

Characteristics that have infinite slope under the first definition (2.293) will have zero slope when defined in this manner.

If a discontinuity exists, it can only move with speeds given by the characteristics. This calculation thus "uncovers" some property of the system. Specifically, it reveals the directions in which information about the solution travels over time. As before, a bit of information travels along each characteristic; here that information is that a discontinuity might exist.

2.4.12 Discontinuity traces in more spatial dimensions

Suppose now that the dependent variables are distributed over more than one spatial dimension. If a discontinuity exists in a solution that is distributed over n dimensions, it will be across a surface of at most $n-1$ dimensions. In figure 2-7, the domain is a line and the discontinuity, here in the first derivative of the solution with respect to x, exists across a point. In figure 2-8, the domain is a plane and the discontinuity, here in the value of the solution itself, exists across a line.

The multidimensional analog of discontinuity traces is very straightforward. Consider a system of partial differential equations in n spatial dimensions.

$$\mathbf{A}\mathbf{u}_t + \mathbf{B}_1\mathbf{u}_{x_1} + \mathbf{B}_2\mathbf{u}_{x_2} + \cdots + \mathbf{B}_n\mathbf{u}_{x_n} = \mathbf{f} \tag{2.296}$$

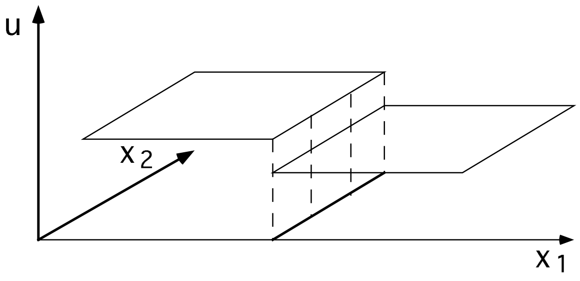

Figure 2-8: Discontinuous solution in two dimensions

and assume that Cauchy data for the system is known on some surface in $n-1$ spatial dimensions and time. At any (smooth) point, the surface will be defined by its normal at that point, and will have $n-1$ basis vectors tangent to it at that point. Since $\mathbf{u}$ is known over the entire surface, all interior partial derivatives are also known at that point. The problem is again to determine the conditions under which a discontinuity might exist across the surface at that point.

The first step is to split the partial differential operators of the original system (2.296) into their interior and exterior components. If $\bar{x}$ is the coordinate along $\mathbf{p}$, the normal or exterior direction, then

$$\frac{\partial}{\partial x_i} = \frac{\partial \bar{x}}{\partial x_i}\frac{\partial}{\partial \bar{x}} + \text{interior components} \tag{2.297}$$

x_i is the distance along the i^{th} coordinate vector $\mathbf{x}_i$, which is a vector $\mathbf{z}$ with $z_i = 1$ and $z_{j\neq i} = 0$. $\bar{x}$ is then related to x_i by the projection of $\mathbf{x}_i$ onto $\mathbf{p}$.

$$\bar{x}\left(x_i\right) = \frac{\mathbf{x}_i \cdot \mathbf{p}}{\mathbf{p}\cdot\mathbf{p}} \tag{2.298}$$

so, for unit $\mathbf{p}$,

$$\frac{\partial \bar{x}}{\partial x_i} = p_i \tag{2.299}$$

Next, one can use the transformed derivatives (2.297) to replace all partial derivatives in the original equations (2.296) with their interior and exterior components. Since all interior components of the derivatives are known, they may be moved to the righthand side of the equation and included in a new forcing term $\mathbf{g}$. This leaves

$$\mathbf{A}\mathbf{u}_t + \left[\mathbf{B}_1\frac{\partial \bar{x}}{\partial x_1} + \mathbf{B}_2\frac{\partial \bar{x}}{\partial x_2} + \cdots + \mathbf{B}_n\frac{\partial \bar{x}}{\partial x_n}\right]\mathbf{u}_{\bar{x}} = \mathbf{g} \tag{2.300}$$

which, by using (2.299), reduces to

$$\mathbf{u}_t + \mathbf{B}\mathbf{u}_{\bar{x}} = \mathbf{g} \tag{2.301}$$

where

$$\mathbf{B} = \sum_{i=1}^{n} p_i \mathbf{B}_i \tag{2.302}$$

This (2.301) is called the **projected system** [17].

Now, for a discontinuity to exist across the surface at the point under consideration, this system of equations must be insufficient to determine the derivatives in the $\bar{x}$ direction. Proceeding in precisely the same manner as in the one dimensional case, let λ be the speed in the $(\bar{x}, t)$ plane with which the Cauchy data travels. A discontinuity can exist only if the projected system (2.301) does not uniquely determine the exterior partial derivatives $\mathbf{u}_{\bar{x}}$. This means that the speed λ must satisfy the characteristic condition

$$\Big|\mathbf{B} - \lambda \mathbf{I}\Big| = 0 \tag{2.303}$$

A projected system allows one to determine proper boundary conditions for partial differential equations on multidimensional domains. If $\mathbf{p}$ is chosen as the unit outward normal to the domain at some point of interest, then the characteristics of the projected system will determine how many boundary conditions are required. Every negative eigenvalue of the characteristic condition for the projected system corresponds to a characteristic directed into the domain. As before, for every such characteristic travelling into the domain, a boundary condition will be required [18]. One can then transform the projected equations to their characteristic form as in the one-dimensional case, and identify the subspace that the boundary conditions must span.

Chapter 3

The Differentiation Index of a PDE

3.1 Introduction

Automated index analysis of general DAEs has proven extremely useful in process simulators [26]. In particular, Pantelides' algorithm allows a process simulator to efficiently estimate the differentiation index of large, nonlinear dynamic models. Using the information provided by Pantelides' algorithm, a process simulator can go one step further and generate a mathematically equivalent low-index reformulation [56] that is suitable for numeric solution. This allows an engineer who has no knowledge of index analysis to formulate a high index process model and use it for dynamic simulation. It is also required for automatic solution of a broad class of constrained dynamic optimizations [26].

No comparable analysis exists for dynamic flowsheet simulations that are based on distributed unit models, because no definition of an index for partial differential equations upon which such an analysis may be constructed has previously been developed. This chapter will present a new approach to index analysis of partial differential equations that is built from a very natural generalization of the differentiation index of differential-algebraic systems. As such, it allows many of the algorithms and analyses that have proven valuable in the case of lumped model formulations to be applied with minimal modification to distributed model formulations as well.

Previous approaches to index analysis of partial differential equations have focused on a perturbation and an algebraic index. Campbell and Marszalek [12] have defined a perturbation index for parabolic linear systems of the form

$$\begin{gathered}\mathbf{A}\mathbf{u}_t + \mathbf{B}\mathbf{u}_{xx} + \mathbf{C}\mathbf{u}_x + \mathbf{D}\mathbf{u} = \mathbf{f}(t,x) \\ 0 \le x \le L, \quad 0 \le t \le T \\ \mathbf{u}(0,t) = 0 \quad \mathbf{u}(L,t) = 0 \\ \mathbf{u}(x,0) = \mathbf{u}_0(x)\end{gathered} \tag{3.1}$$

They consider only solutions that identically satisfy the boundary data, specifically sine series.

$$\mathbf{u}(x,t) = L^{-\frac{1}{2}} \sum_{j=1}^{\infty} \mathbf{u}_j(t) \sin(\frac{j\pi x}{L}) \tag{3.2}$$

It is assumed that the data $\mathbf{u}_0(x)$ and forcing functions $\mathbf{f}(t,x)$ may also be represented as sine series, so

$$\mathbf{u}_0(x) = L^{-\frac{1}{2}} \sum_{j=1}^{\infty} \mathbf{u}_{0j} \sin(\frac{j\pi x}{L}) \tag{3.3}$$

$$\mathbf{f}(t,x) = L^{-\frac{1}{2}} \sum_{j=1}^{\infty} \mathbf{f}_j(t) \sin(\frac{j\pi x}{L}) \tag{3.4}$$

If $||\cdot||$ is the Euclidean norm on $\mathbb{R}^n$ and $||\cdot||_2$ is the L_2 norm in the x variable, $||\mathbf{c}(t,x)||_\infty$ is defined as

$$||\mathbf{c}(t,x)||_\infty = \max_{0\le t\le T} \left(\int_0^L ||\mathbf{c}(t,x)||^2 dx \right)^{\frac{1}{2}} = \max_{0\le t\le T} ||\mathbf{c}(t,x)||_2 \tag{3.5}$$

Then $||\cdot||_{(p,q)}$ is defined as

$$||\mathbf{c}(t,x)||_{(p,q)} = \sum_{i=0}^{p} \sum_{k=0}^{q} ||\frac{\partial^{i+k}}{\partial t^i \partial x^k} \mathbf{c}(t,x)||_\infty \tag{3.6}$$

Let the solution $\mathbf{u}(t,x)$ satisfy (3.1) for some $\mathbf{f}(t,x)$ and associated consistent $\mathbf{u}_0(x)$. The **infinite perturbation index** ν_p^∞ is defined as

$$\begin{aligned}\nu_p^\infty = 1 + \min \Big(& \max(p_1{+}q_1, q_2) : \\ & ||\hat{\mathbf{u}} - \mathbf{u}|| \le C_1 ||\hat{\mathbf{f}} - \mathbf{f}||_{(p_1,q_1)} + C_2 ||\hat{\mathbf{u}}_0 - \mathbf{u}_0||_{(0,q_2)} \Big)\end{aligned} \tag{3.7}$$

where $\hat{\mathbf{u}}(t,x)$ is some other solution that satisfies (3.1) for some $\hat{\mathbf{f}}(t,x)$ in a neigborhood of $\mathbf{f}(t,x)$ and associated consistent $\hat{\mathbf{u}}_0(x)$. The **maximum perturbation index** ν_P^∞ is then defined as the maximum of ν_p^∞ over a neighborhood of $\mathbf{u}$.

The perturbation index is calculated from the solution to the original problem, given a decision on what data (if any) is used to restrict the solution and the analytic form of that solution in terms of the remaining data. It may be defined for nonlinear systems in a similar manner. Here it is assumed that all boundary conditions are used to restrict the solution, so they are not included in the index.

Günther and Wagner [36] consider instead solutions of linear hyperbolic systems of first or second order. This necessitates modification of the definition of the perturbation index. They define an infinite perturbation index that includes perturbations of the boundary data $\mathbf{s}(t)$.

$$\begin{aligned}\nu_p^\infty = 1 + \min\Big(& \max(p_1 + q_1, p_2, q_2) : \\ & ||\hat{\mathbf{u}} - \mathbf{u}|| \leq C_1 ||\hat{\mathbf{f}} - \mathbf{f}||_{(p_1,q_1)} + C_2 ||\hat{\mathbf{u}}_0 - \mathbf{u}_0||_{(0,q_2)} + C_3 ||\hat{\mathbf{s}} - \mathbf{s}||_{(p_2,0)} \Big)\end{aligned} \tag{3.8}$$

Similarly, the (maximum) perturbation index ν_P^∞ is then defined as the maximum of ν_p^∞ in a neighborhood of $\mathbf{u}$.

These perturbation indices capture the dependence of a solution to a PDE on derivatives of both the forcing functions and data. This dependence on derivatives of the data may be function of how the data is specified [12].

For application in a process simulator, the perturbation index approach has several shortcomings. First, the perturbation index is a property of the analytical solution. As such, it is unsuitable for *a priori* analysis of general models for which the analytical solution may not be available. Second, it assumes that proper initial and boundary data are known, and therefore cannot be used to guide the user in specification of data. Third, it requires a decision regarding whether or not each datum is to be used to restrict the function space from which the solution is constructed.

Several **algebraic indices** of a linear PDE are defined by Campbell and Marszalek [12] for the algebraic system that results from solving the original PDE in the Laplace domain. For example, given a system of the form

$$\mathbf{A}\mathbf{u}_t + \mathbf{D}\mathbf{u}_{xx} + \mathbf{B}\mathbf{u}_x - \mathbf{C}\mathbf{u} = \mathbf{f}(t, x) \tag{3.9}$$

with $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in \mathbb{R}^{n \times n}$, the resolvent $\mathbf{R}(s, z)$ is

$$\mathbf{R}(s, z) = (s\mathbf{A} + z^2\mathbf{D} + z\mathbf{B} - \mathbf{C})^{-1} \tag{3.10}$$

$\mathbf{R}$ is a matrix of rational functions in the real variables s, z. Recall that a quotient $r(s, z)$ of two real polynomials in the real variables s, z is said to be s-proper if $\lim_{|s|\to\infty} r(s, z) = 0$ for almost all z, and that a matrix is s-proper if every one of its entries is s-proper. The **algebraic t-index** is then defined as the smallest integer n_1 such that $s^{-n_1}\mathbf{R}(s, z)$ is s-proper, and the **algebraic x-index** is similarly defined as the smallest integer n_2 such that $z^{-n_2}\mathbf{R}(s, z)$ is z-proper.

A quotient $r(s, z)$ is said to be proper if it is both s-proper and z-proper. The **algebraic index** ν_A^∞ is then defined as

$$\nu_A^\infty = \max_{i,j} \left(\min_{n_1, n_2 \geq 0} (n_1 + n_2 : s^{n_1} z^{n_2} R_{ij}(s, z) \text{ is proper}) \right) \tag{3.11}$$

The algebraic index is a property of the governing equations themselves, and not of specific values of data or domain geometry. It is therefore independent of whether the solution is restricted or unrestricted, and thus closer to the type of analysis that would be suitable for a process simulator. However, it is not defined for general nonlinear systems.

In order to address the issue of guiding the user in the specification of data in both the linear and nonlinear case, this work develops an index by focusing instead on Cauchy data. Recall that Cauchy data are the values of the dependent variables and the exterior derivatives of the variables over an entire hyperplane. Cauchy data represents the generalization of initial data for DAEs to the multidimensional case.

3.2 Defining the differentiation index of a PDE

Consider a first order PDE system over $\mathbb{R}^n$. Call the independent variables $\mathbf{x} \in \mathbb{R}^n$, let the dependent variables be $\mathbf{u} \in \mathbb{R}^m$, and suppose the following PDE holds over the rectangular domain $x_i \in I_i,\ \ i = 1 \dots n$.

$$\mathbf{F}\left(\mathbf{u}_{x_{i=1\dots n}}, \mathbf{u}, \mathbf{x}\right) = 0 \tag{3.12}$$

Here $\mathbf{u}_{x_i} = \frac{\partial \mathbf{u}}{\partial x_i} \in \mathbb{R}^m, \mathbf{u} \in \mathbb{R}^m, \mathbf{x} \in \mathbb{R}^n$, $\mathbf{F} : \mathbb{R}^{2m+n} \rightarrow \mathbb{R}^m$, and I_i is a subinterval of $\mathbb{R}$, for $i = 1 \dots n$; each interval has the form $a_i \leq x_i \leq b_i$ for some real constants a_i and b_i. Existence of the solution $\mathbf{u}$ is assumed.

In order to make the parallel with the DAE case more clear, denote the partial derivative of $\mathbf{u}$ with respect to x_j by a dot, so $\mathbf{u}_{x_j} = \dot{\mathbf{u}}$. For all other $i \neq j$, partial differentiation will still be denoted by a subscripted independent variable. The dot denotes differentiation with respect to the direction exterior to the hyperplane $x_j =$ *constant*; all other partial derivatives are interior on that hyperplane. Using this notation, the general system (3.12) is written as

$$\mathbf{F}\left(\dot{\mathbf{u}}, \mathbf{u}_{x_{i=1\dots n, i\neq j}}, \mathbf{u}, \mathbf{x}\right) = 0 \tag{3.13}$$

Note that $\mathbf{J}(\mathbf{F}, \dot{\mathbf{u}})$, the Jacobian of $\mathbf{F}$ with respect to $\dot{\mathbf{u}}$, may be singular. Under the assumptions that a solution $\mathbf{u}$ exists and that $\mathbf{F}$ is sufficiently differentiable, the differentiation index of this PDE may be defined as follows.

Definition 3.2.1 *The* **differentiation index with respect to** x_j*, or* ν_{x_j}*, is the smallest number of times that some or all of the elements of* $\mathbf{F}$ *must be differentiated with respect to* x_j *in order to determine* $\dot{\mathbf{u}}$ *as a continuous function of* $\mathbf{u}_{x_{i\neq j}}$*,* $\mathbf{u}$*, and* $\mathbf{x}$*.*

A formal index analysis built on the concept of a derivative array for PDEs may be most easily constructed for linear systems. Such an analysis is not as straightforward as that for linear DAEs, however, because one must consider operator-valued coefficient matrices. This analysis may be extended fairly readily to a particular class of semilinear systems, of which linear systems are a special case, so this formal index analysis will be presented only once, for the more general class of systems.

Consider a PDE system of the following form.

$$\mathbf{F}\left(\dot{\mathbf{u}}, \mathbf{u}_{x_{i=1\dots n, i\neq j}}, \mathbf{u}, \mathbf{x}\right) = \sum_{i=1}^{n} \mathbf{A}_i(x_j)\mathbf{u}_{x_i} + \mathbf{C}(x_j)\mathbf{u} - \mathbf{f}(\mathbf{x}) = 0 \tag{3.14}$$

$\mathbf{A}_i(x_j), \mathbf{C}(x_j) : \mathbb{R}^1 \to \mathbb{R}^{m\times m}$, and all other quantities are defined as in the general case (3.12). Such a system will hereafter be referred to as a linear x_j-varying PDE.

The system may be rewritten as

$$\mathbf{A}(x_j)\dot{\mathbf{u}} + \mathbf{B}(x_j)\mathbf{u} - \mathbf{f}(\mathbf{x}) = 0 \tag{3.15}$$

where

$$\begin{aligned} \mathbf{A}(x_j) &= \mathbf{A}_j(x_j) \\ \dot{\mathbf{u}} &= \mathbf{u}_{x_j} \\ \mathbf{B}(x_j) &= \mathbf{C}(x_j) + \sum_{i\neq j} \mathbf{A}_i(x_j) D_{x_i} \\ D_{x_i} &= \frac{\partial}{\partial x_i} \end{aligned}$$

This system (3.15) has the same form as a linear time-varying DAE.

However, here $\mathbf{B} \in P_{Ij}^{m\times m}$, the set of all m by m matrices whose elements belong to P_{Ij}. $P_{Ij} = \{L \mid Lu = \sum_{\boldsymbol{\tau}} l_{\boldsymbol{\tau}}(x_j) D_{\boldsymbol{\tau}} u, u \in \mathbb{R}\}$, where $\boldsymbol{\tau} \in \mathbb{Z}^n$ is a multi-index with $\tau_i \in \mathbb{Z}^{+^n}$ and $\tau_j = 0$; $l_{\boldsymbol{\tau}}(x_j) : \mathbb{R}^1 \to \mathbb{R}^1$ and is analytic for $x_j \in I_j$, I_j is a closed interval in $\mathbb{R}$, and $D_{\boldsymbol{\tau}} = \prod_{i=1}^n \left(\frac{\partial}{\partial x_i}\right)^{\tau_i}$. P_{Ij} is the set of all interior partial differential operators on any hyperplane ϕ orthogonal to x_j given by $x_j = c$, $c \in I_j$, with coefficients that vary smoothly in x_j over I_j. Any $p \in P_{Ij}$ is a linear operator on ϕ.

The operators + and × are defined as follows, for any two operators $a, b \in P_{Ij}$.

$$a + b = \sum_{\boldsymbol{\nu}} a_{\boldsymbol{\nu}} D_{\boldsymbol{\nu}} + \sum_{\boldsymbol{\nu}} b_{\boldsymbol{\nu}} D_{\boldsymbol{\nu}} = \sum_{\boldsymbol{\nu}} \left(a_{\boldsymbol{\nu}} + b_{\boldsymbol{\nu}}\right) D_{\boldsymbol{\nu}}$$

$$a \times b = \left(\sum_{\boldsymbol{\nu}} a_{\boldsymbol{\nu}} D_{\boldsymbol{\nu}}\right) \left(\sum_{\boldsymbol{\gamma}} b_{\boldsymbol{\gamma}} D_{\boldsymbol{\gamma}}\right) = \sum_{\boldsymbol{\nu}} \sum_{\boldsymbol{\gamma}} a_{\boldsymbol{\nu}} b_{\boldsymbol{\gamma}} D_{\boldsymbol{\nu}+\boldsymbol{\gamma}}$$

Lemma 3.2.2 $\left\langle P_{Ij}^{m\times m}, +, \times \right\rangle$ *is a ring.*

Proof. $\langle P_{Ij}, + \rangle$ is an abelian group. × is associative on P_{Ij} and is left and right distributive with +. Therefore $\langle P_{Ij}, +, \times \rangle$ is a ring. The set $P_{Ij}^{m\times m}$ of all square matrices whose elements belong to P_{Ij} forms a ring with the same operators [29]; thus $\left\langle P_{Ij}^{m\times m}, +, \times \right\rangle$ is a ring. □

Thus, many results from standard matrix algebra also hold for $P_{Ij}^{m\times m}$. For example, row operations may be used to permute rows, scale or add rows together, perform Gauss elimination, and evaluate determinants.

Lemma 3.2.3 *For* $\mathbf{A} \in P_{Ij}^{m\times m}$*, if* $|\mathbf{A}| \neq 0$*, then* $\exists\ \mathbf{R} \in P_{Ij}^{m\times m}$ *such that* $\mathbf{RA} = \mathbf{D}$*, where* $\mathbf{D} \in P_{Ij}^{m\times m}$ *is a diagonal matrix with* $d_{ii} \neq 0$*.*

Proof. Because $\left\langle P_{Ij}^{m\times m}, +, \times \right\rangle$ is a ring, Gauss elimination may be used to produce first an upper triangular and then a diagonal matrix through row operations alone. Therefore, Gauss elimination gives a sequence of row operations $\mathbf{R} \in P_{Ij}^{m\times m}$ for which $\mathbf{RA} = \mathbf{D}$. If $|\mathbf{A}| \neq 0$, the elements of the diagonal matrix $\mathbf{D}$ will be strictly nonzero. □

Example 1 *Consider a matrix* $\mathbf{A} \in P_{I4}^{2\times 2}$*, with* $n = 4$ *and* $I_j = \{x_j \mid 1 \leq x_j \leq 10\}$*.*

$$\mathbf{A} = \begin{bmatrix} 3x_4^2 & \frac{\partial^2}{\partial x_1 \partial x_2} \\ 2\frac{\partial}{\partial x_1} + 5\frac{\partial}{\partial x_3} & 7x_4 \end{bmatrix} \tag{3.16}$$

$\mathbf{A}$ *is nonsingular, because*

$$|\mathbf{A}| = 21x_4^3 - 2\frac{\partial^3}{\partial x_1^2 \partial x_2} - 5\frac{\partial^3}{\partial x_1 \partial x_2 \partial x_3} \neq 0 \tag{3.17}$$

If a series of row operations defined as a matrix $\mathbf{R} \in P_{I4}^{2\times 2}$ *are given by*

$$\mathbf{R} = \begin{bmatrix} -21x_4^3 & 3x_4^2\frac{\partial^2}{\partial x_1 \partial x_2} \\ 2\frac{\partial}{\partial x_1} + 5\frac{\partial}{\partial x_3} & -3x_4^2 \end{bmatrix} \tag{3.18}$$

then $\mathbf{RA}$ *is a diagonal matrix.*

$$\mathbf{RA} = \begin{bmatrix} -63x_4^5 + 6x_4^2\frac{\partial^3}{\partial x_1^2 \partial x_2} + 15x_4^2\frac{\partial^3}{\partial x_1 \partial x_2 \partial x_3} & 0 \\ 0 & 21x_4^3 + 2\frac{\partial^3}{\partial x_1^2 \partial x_2} + 5\frac{\partial^3}{\partial x_1 \partial x_2 \partial x_3} \end{bmatrix} \tag{3.19}$$

In analogy with DAEs, the derivative array equations for the PDE (3.15) may be derived up to any order of differentiation with respect to x_j, as long as $\mathbf{A}(x_j)$, $\mathbf{B}(x_j)$, $\mathbf{f}$, and $\mathbf{u}$ are sufficiently differentiable. In the linear case, $\mathbf{A}^{(l)} = \mathbf{B}^{(l)} = \mathbf{0}$ if $l > 0$, so the derivative array equations with respect to x_j are

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{B} & \mathbf{A} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{B} & \mathbf{A} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(1)} \\ \cdot \\ \cdot \\ \mathbf{u}^{(k)} \end{bmatrix} = - \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \cdot \\ \cdot \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{f}^{(0)} \\ \cdot \\ \cdot \\ \mathbf{f}^{k-1} \end{bmatrix} \tag{3.20}$$

or

$$\mathcal{A}_k \mathbf{u}_k = -\mathcal{B}_k \mathbf{u} + \mathbf{f}_k \tag{3.21}$$

The following result for linear PDEs (3.14) will be useful later.

Theorem 3.2.4 $\nu_{x_i} \geq 1$ *iff* $|\mathbf{A}_i| = 0$.

Proof. If $\nu_{x_i} \geq 1$, then by the definition of the differentiation index, the system does not uniquely determine $\mathbf{u}_{x_i}$, and thus $|\mathbf{J}\left(\mathbf{F}, \mathbf{u}_{x_i}\right)| = 0$. Since $\mathbf{J}\left(\mathbf{F}, \mathbf{u}_{x_i}\right) = \mathbf{A}_i$, then $|\mathbf{A}_i| = 0$. Similarly, if $|\mathbf{A}_i| = 0$, then $|\mathbf{J}\left(\mathbf{F}, \mathbf{u}_{x_i}\right)| = 0$ and the system cannot be solved for unique $\mathbf{u}_{x_i}$, and by definition $\nu_{x_i} \geq 1$. □

In the linear x_j-varying case, the first k derivative array equations with respect to x_j are

$$\begin{bmatrix} \mathbf{A}^{(0)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}^{(1)} + \mathbf{B}^{(0)} & \mathbf{A}^{(0)} & \dots & \mathbf{0} \\ \mathbf{A}^{(2)} + 2\mathbf{B}^{(1)} & 2\mathbf{A}^{(1)} + \mathbf{B}^{(0)} & \ddots & \mathbf{0} \\ \vdots & \vdots & & \mathbf{A}^{(0)} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \\ \vdots \\ \mathbf{u}^{(k)} \end{bmatrix} = - \begin{bmatrix} \mathbf{B}^{(0)} \\ \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{B}^{(k-1)} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{f}^{(0)} \\ \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(k-1)} \end{bmatrix} \tag{3.22}$$

While the derivative array equations have the same form as given for linear time-varying DAEs, $\mathbf{A}^{(i)} = \left(\frac{\partial}{\partial x_j}\right)^i \mathbf{A} \in P_{Ij}^{m \times m}$ and $\mathbf{B}^{(i)} = \left(\frac{\partial}{\partial x_j}\right)^i \mathbf{B} \in P_{Ij}^{m \times m}$.

The ring property of $P_{Ij}^{m \times m}$ allows the following definition.

Definition 3.2.5 *The matrix $\mathcal{A}_k$ is smoothly 1-full on ϕ_{Ij} if there is a smooth non-singular $\mathbf{R}(x_j)$ such that*

$$\mathbf{R}\mathcal{A}_k = \begin{bmatrix} \mathbf{D}(x_j) & \mathbf{0} \\ \mathbf{0} & \mathbf{H}(x_j) \end{bmatrix}$$

where $\mathbf{D}(x_j) \in P_j^{m \times m}$ *is a nonsingular diagonal matrix and* ϕ_{Ij} *is the set of hyperrectangles orthogonal to the* x_j *coordinate direction given by* $\{\mathbf{x} \mid x_j = c, c \in I_j; x_{i=1\ldots n, i \neq j} \in I_i\}$*.*

When $\mathcal{A}_k$ is smoothly 1-full on ϕ_{Ij}, the $k-1$ differentiations with respect to x_j that generate the derivative array equations give $\dot{\mathbf{u}}$ as a continuous function of $\mathbf{u}$ and $\mathbf{x}$ over $\phi \in \phi_{Ij}$. As with DAEs, the solution of an index ν_{x_j} linear or linear x_j-varying PDE will depend *explicitly* on up to $\nu_{x_j} - 1$ derivatives with respect to x_j of the forcing functions over ϕ_{Ij}. As will be shown later, there may also be *implicit* dependence on derivatives of the forcing functions .

Theorem 3.2.6 *For a linear* x_j*-varying system, if* k *is the smallest integer such that* $\mathcal{A}_k$ *is smoothly 1-full over* I_j*, the maximum index* ν_{x_j} *on* ϕ_{Ij} *is* $k-1$*.*

Proof. Suppose that, on some hyperrectangle $x_j = c$, $c \in I_j$, the index ν_{x_j} is greater than $k-1$. Then $k-1$ differentiations do not determine $\dot{\mathbf{u}}$. But $\mathcal{A}_k$ is smoothly 1-full on I_j, so it does determine $\dot{\mathbf{u}}$ as a continuous function at $x_j = c$, and the index cannot be greater than $k-1$. □

For more general semilinear and nonlinear systems, the derivative array equations may still be defined, again provided that the original system is sufficiently differentiable. However, they may not have the convenient matrix structure that exists for linear and linear x_j-varying systems. Even for simple linear systems, the full derivative array equations are often not calculated, as only a subset of the equations constrain Cauchy data when differentiated[1]. Furthermore, nonlinear systems may develop discontinuous solutions even given smooth data and forcing functions. For such systems, the index is therefore a *local* property in $(\mathbf{u}, \mathbf{x})$-space, just as the differentiation index is a local property for nonlinear DAEs.

[1] In the following examples, typically only this subset of the equations will be differentiated.

3.3 Consistent Cauchy data and the differentiation index

Suppose Cauchy data is to be specified on a hyperplane orthogonal to the x_j coordinate direction given by $x_j = x_{j0} \in I_j$. Cauchy data on this surface is the values of $\dot{\mathbf{u}}_0$ and $\mathbf{u}_0$ over the entire surface. In order for this data to be consistent with the original equation (3.12), clearly it must satisfy

$$\mathbf{F}\left(\dot{\mathbf{u}}_0, \mathbf{u}_{0x_{i\neq j}}, \mathbf{u}_0, x_{i\neq j}, x_{j0}\right) = 0 \tag{3.23}$$

Determination of ν_{x_j} will derive any other equations that restrict consistent Cauchy data, in a manner similar to how determination of the index of a DAE uncovers the complete set of equations that must be satisfied by consistent initial conditions.

Example 2 *Consider the following system.*

$$\begin{aligned} u_{x_1} - v_{x_2} &= 0 \\ v_{x_1} - u_{x_2} &= 0 \end{aligned} \tag{3.24}$$

over $0 \leq x_1, a \leq x_2 \leq b$*. Suppose one wants to specify Cauchy data on the hyperplane given by* $x_1 = 0$*. Clearly such data must satisfy the original equations, rewritten using a dot to again denote differentiation along the exterior direction.*

$$\begin{aligned} \dot{u} - v_{x_2} &= 0 \\ \dot{v} - u_{x_2} &= 0 \end{aligned} \tag{3.25}$$

on $(x_1 = 0)$*. No additional independent equations relating* $\dot{u}$ *and* $\dot{v}$ *may be derived through differentiation with respect to* x_1*; the system determines* $\dot{u}$ *and* $\dot{v}$*, so its index with respect to* x_1 *is 0. Two degrees of freedom exist for specification of Cauchy data on* $(x_1 = 0)$*.*

This system is the wave equation, written as a first order system. The question of Cauchy data on $(x_1 = 0)$ corresponds to the initial conditions. For the wave equation, initial conditions are typically provided as values of u and v over the initial hyperplane. Alternative specifications of Cauchy data, involving ordinary differential or partial differential equations, will be considered in the next section.

Example 3 *Consider the following system.*

$$\begin{aligned} u_{x_1} - v &= 0 \\ u_{x_2} &= f_1(x_1) \end{aligned} \tag{3.26}$$

Suppose one wishes to specify Cauchy data on $(x_1 = 0)$. *Such data must of course satisfy*

$$\begin{aligned} \dot{u} - v &= 0 \\ u_{x_2} &= f_1(x_1) \end{aligned} \tag{3.27}$$

Differentiating the second equation with respect to x_1 *produces another independent equation involving* $\dot{u}$.

$$\dot{u}_{x_2} = \frac{d}{dx_1} f_1(x_1) \tag{3.28}$$

Differentiating this new equation and the first equation in (3.27) gives two additional equations in the two new unknowns $\ddot{u}$ *and* $\dot{v}$.

$$\begin{aligned} \ddot{u} - \dot{v} &= 0 \\ \ddot{u}_{x_2} &= \frac{d^2}{dx_1^2} f_1(x_1) \end{aligned} \tag{3.29}$$

Assuming that $f_1(x_1)$ *is twice continuously differentiable, two differentiations with respect to* x_1 *give* u_{x_1} *and* v_{x_1} *as continuous functions of* u, v, *and* $\mathbf{x}$; *thus* ν_{x_1}, *the index of this system with respect to* x_1, *is 2. The system (3.27 - 3.29) is fully determined in the variables* $\ddot{u}, \dot{u}, u, \dot{v}$, *and* v; *no degrees of freedom are available for the specification of Cauchy data on the hyperplane* $(x_1 = 0)$.

Now, suppose one wishes instead to specify Cauchy data on $(x_2 = 0)$. *The exterior direction to this hyperplane is* x_2 *and the system may be rewritten for clarity as*

$$\begin{aligned} u_{x_1} - v &= 0 \\ \dot{u} &= f_1(x_1) \end{aligned} \tag{3.30}$$

Differentiation of the first equation yields

$$\dot{u}_{x_1} - \dot{v} = 0 \tag{3.31}$$

which gives $\dot{u}$ and $\dot{v}$ as functions of u, v, and $\mathbf{x}$. No additional independent equations may be derived that relate the variables $\dot{u}, u, \dot{v}$, and v. Therefore $\nu_{x_2} = 1$ and there is one degree of freedom available for specification of Cauchy data on the hyperplane. Note, however, that neither $\dot{u}$ nor $\dot{v}$ may be specified; only u or v may be set independently on the hyperplane.

To verify the preceding results, note that the solution of the PDE system above on the semi-infinite domain $0 \leq x_1, -a \leq x_2 \leq a$ is determined by a single function $g(x_1)$ specified at some point c on the interval $-a \leq c \leq a$:

$$\begin{aligned} u(x_1, x_2) &= x_2 f_1(x_1) + g(x_1) \\ v(x_1, x_2) &= x_2 \frac{d}{dx_1} \left(f_1(x_1) + g(x_1) \right) \end{aligned} \tag{3.32}$$

Thus the solution on the hyperplane ($x_1 = 0$) is given by $f(0)$ and $g(0)$; no other degrees of freedom remain (as indicated by the index analysis). The conditions for consistency with the equations fully determine all Cauchy data on that hyperplane.

Example 4 *The hyperplane on which Cauchy data is analyzed for consistency with the equations need not be orthogonal to one of the original coordinate axes. If the index with respect to a non-coordinate direction is needed, the coordinates may be transformed so that one of the new coordinate vectors lies along the direction of interest, and all other coordinate vectors are orthogonal to the direction of interest.*

Consider the index of the one-way wave equation

$$cu_{x_1} + u_{x_2} = 0 \tag{3.33}$$

with respect to the direction $(x_1, x_2) = (1, -1)$. Define a new coordinate system by

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.34}$$

so that the direction of interest is now in the y direction, and the z direction is orthogonal to the y direction. Transforming to the new coordinate system, the equation becomes

$$(c-1)\dot{u} + (c+1)u_z = 0 \tag{3.35}$$

The index with respect to y is zero, unless $c = 1$. In this case, the index becomes 1, and Cauchy data on $y = y_0$ must also satisfy

$$\dot{u}_z = 0 \tag{3.36}$$

When $c \neq 1$, the index is zero and either $\dot{u}$ or u may be specified arbitrarily on the hyperplane. In the case $c = 1$, neither may be specified independently. However, there is a **lower dimensionality degree of freedom***. That is, the system (3.35-3.36) constrains both $\dot{u}$ and u over the hyperplane $(y = y_0)$, so that neither may be specified arbitrarily over the entire surface. The value of each may be given arbitrarily at a single point on the surface, and that value determines the Cauchy data.*

Note that the surface $(y = y_0)$ is a characteristic surface of the one-way wave equation when $c = 0$. The differentiation index and characteristics are related by the following theorem.

Theorem 3.3.1 *A hyperplane $\phi(\mathbf{x}) = 0$ is a characteristic surface of a linear, first order PDE system iff $\nu_{\nabla\phi} \geq 1$.*

Proof. Consider a general linear PDE system of first order

$$\sum_{i=1}^{n} \mathbf{A}_i \mathbf{u}_{x_i} = \mathbf{f}(\mathbf{u}, \mathbf{x}) \tag{3.37}$$

and a hyperplane given by

$$\phi\left(\mathbf{x}\right) = 0, \quad \phi_{\mathbf{x}} = [\phi_{x_1}\, \phi_{x_2}\, \ldots\, \phi_{x_n}] \neq 0 \tag{3.38}$$

Consider a coordinate change from $\mathbf{x}$ to $\mathbf{z}$, where $z_n = \phi(\mathbf{x})$, and z_i, $i = 1 \ldots (n-1)$, denotes distance in the direction of basis vector $\mathbf{b}_i$. Let the basis vectors for the new coordinate system be orthogonal, so that $\mathbf{b}_i \cdot \nabla\phi = \mathbf{b}_i \cdot \mathbf{b}_{j \neq i} = 0$. In the new coordinates, the system becomes

$$\mathbf{B}_n \mathbf{u}_{z_n} + \sum_{i=1}^{n-1} \mathbf{B}_i \mathbf{u}_{z_i} = \mathbf{f}(\mathbf{u}, \mathbf{x}(\mathbf{z})) \tag{3.39}$$

where

$$\begin{aligned} \mathbf{B}_n &= \sum_{j=1}^{n} \mathbf{A}_i \phi_{x_i} \\ \mathbf{B}_i &= \sum_{j=1}^{n} \mathbf{A}_j \frac{\partial z_i}{\partial x_j}, \quad i = 1 \ldots n-1 \end{aligned} \tag{3.40}$$

Note that $\mathbf{u}_{z_n}$ is the exterior derivative, and all other derivatives are interior on the surface.

If the system has $\nu_{\nabla\phi} \geq 1$, then by theorem 3.2.4 $|\mathbf{B}_n| = 0$. If $|\mathbf{B}_n| = 0$, the surface is a characteristic surface [59]. Conversely, if the surface is characteristic, then by definition $|\mathbf{B}_n| = 0$ and thus $\nu_{\nabla\phi} \geq 1$. □

Remark: For a system that includes one or more algebraic equations, every hyperplane is a characteristic surface.

3.4 Dynamic degrees of freedom

Index analysis of PDEs with respect to a direction x_j requires derivation of all independent equations that constrain allowable Cauchy data on a hyperplane ϕ orthogonal to that direction. The consistent Cauchy data problem for the original first-order PDE (3.13) consists in part of these m equations in the $2m$ variables $\dot{\mathbf{u}}$ and $\mathbf{u}$. Again note that interior derivatives of $\mathbf{u}$ on ϕ and the value of $\mathbf{u}$ itself over ϕ are not independent; the interior derivatives of $\mathbf{u}$ are simply functions of $\mathbf{u}$ on ϕ.

For a linear system (3.15) with constant coefficient matrices, assume that $(\mathbf{A}, \mathbf{B})$ form a regular pencil, so that $[\mathbf{A} : \mathbf{B}]$ has full rank. This implies that the coefficients in the derivative array equations $[\mathcal{A}_k : \mathcal{B}_k]$ have full rank for any k. Because

$$\mathcal{A}_k \mathbf{u}_k = \mathcal{B}_k \mathbf{u} + \mathcal{F}_k \tag{3.41}$$

has the form

$$\begin{bmatrix} \mathbf{A} & & & \\ \mathbf{B} & \mathbf{A} & & \\ & \ddots & \ddots & \\ & & \mathbf{B} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \ddot{\mathbf{u}} \\ \dddot{\mathbf{u}} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{f} \\ \mathbf{f}' \\ \mathbf{f}'' \\ \vdots \end{bmatrix} \tag{3.42}$$

then $[\mathcal{A}_k : \mathcal{B}_k]$, which is simply

$$\begin{bmatrix} \mathbf{A} & & & & : & \mathbf{B} \\ \mathbf{B} & \mathbf{A} & & & : & \\ & \ddots & \ddots & & : & \\ & & \mathbf{B} & \mathbf{A} & : & \end{bmatrix} \tag{3.43}$$

has full row rank.

Now suppose that the index of the system with respect to x_j is $k-1$, so that $\mathcal{A}_k$ is smoothly 1-full. This means that there exists a set of operator-valued row operations $\mathbf{R}_1$ that perform Gauss elimination on $[\mathcal{A}_k : \mathcal{B}_k]$ to produce

$$\mathbf{R}_1[\mathcal{A}_k : \mathcal{B}_k] = \begin{bmatrix} \mathbf{D} & \mathbf{0} & : & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{H} & : & \mathbf{B}_2 \end{bmatrix} \tag{3.44}$$

where $\mathbf{D}, \mathbf{B}_1 \in \mathbb{P}^{m \times m}$, $\mathbf{H} \in \mathbb{P}^{(k-1)m \times (k-1)m}$, and $\mathbf{B}_2 \in \mathbb{P}^{(k-1)m \times m}$.

Gauss elimination does not alter the rank of a matrix, so $\mathbf{R}_1[\mathcal{A}_k : \mathcal{B}_k]$ also has full rank. It is possible, however, that $\mathcal{A}_k$ alone does not have full rank. Let the dimension of the nullspace of $\mathcal{A}_k$ be η, so that further Gauss elimination operations $\mathbf{R}_2$ produce η identically zero rows along the bottom of $\mathcal{A}_k$.

$$\mathbf{R}_2\mathbf{R}_1[\mathcal{A}_k : \mathcal{B}_k] = \begin{bmatrix} \mathbf{D} & \mathbf{0} & : & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{G} & : & \mathbf{B}_3 \\ \mathbf{0} & \mathbf{0} & : & \mathbf{B}_4 \end{bmatrix} \tag{3.45}$$

Here $\mathbf{G} \in \mathbb{P}^{(k-1)m-\eta \times (k-1)m}$, $\mathbf{B}_3 \in \mathbb{P}^{(k-1)m-\eta \times m}$, and $\mathbf{B}_4 \in \mathbb{P}^{\eta \times m}$. Again because Gauss elimination does not alter the rank of the matrix, $\mathbf{B}_4$ must have full row rank.

Going back to the derivative array equations that correspond to the first and third block rows of the matrix produced by Gauss elimination (3.45), it is clear that the values of $\dot{\mathbf{u}}$ and $\mathbf{u}$ are partially determined by the n equations of the first block row

$$\mathbf{D}\dot{\mathbf{u}} = \mathbf{B}_1\mathbf{u} + \mathbf{g}_1 \tag{3.46}$$

and the η equations of the last block row

$$\mathbf{0} = \mathbf{B}_4\mathbf{u} + \mathbf{g}_2 \tag{3.47}$$

where $\mathbf{g}_1$ and $\mathbf{g}_2$ are the first m and last η elements of $\mathbf{R}_2\mathbf{R}_1\mathcal{F}_k$, respectively. These $m+\eta$ equations in the $m+m$ variables $\dot{\mathbf{u}}$ and $\mathbf{u}$ are underspecified if $\eta < m$.

For a semilinear, quasilinear, or nonlinear system, algebraic manipulation of all or a subset of the derivative array equations may be employed on a case-by-case basis to determine what variables are determined by algebraic or interior partial differential equations over ϕ. Note that this analysis does not provide any information regarding the well-posedness of the resulting interior partial differential equations.

For a linear or linear $x_{\bar{j}}$-varying system, let $r = m - \eta$. The definition of dynamic degrees of freedom for DAEs [85] then generalizes naturally to PDAEs.

Definition 3.4.1 *Variables* $\mathbf{u}$ *or their exterior derivatives* $\dot{\mathbf{u}}$ *which can be assigned arbitrary distributions over* ϕ *and still allow solution of (3.13) are called* **dynamic degrees of freedom on** ϕ*;* r *dynamic degrees of freedom on* ϕ *must be specified to fully determine Cauchy data on* ϕ*.*

For nonlinear systems, r may also be determined from the derivative array equations by examining the degrees of freedom available in the set of all dependent variables and their exterior partial derivatives.

3.5 Consistent Cauchy data subproblems

A DAE initialization problem always produces an algebraic system. However, with PDEs, a consistent Cauchy data problem may itself be another PDE, in more dependent variables over one fewer independent variable than the original system. Determination of a unique solution may require additional data in the form of side or boundary conditions.

In the following examples, all equations, forcing functions, and dependent variables are assumed to possess all required partial derivatives.

Example 5 *Consider the equations that consistent Cauchy data must obey on a characteristic manifold of the one-way wave equation (3.35 - 3.36). With two equations in* u *and* $\dot{u}$*, there are no dynamic degrees of freedom on* $(y = y_0)$*.*

Let $p = u$ and $q = \dot{u}$, so that the consistent Cauchy data problem is written as

$$\begin{aligned} p_z &= 0 \\ q_z &= 0 \end{aligned} \tag{3.48}$$

For the original system of one dependent variable over two independent variables, our consistent Cauchy data problem is a system of two dependent variables over a single independent variable.

It is a simple, index-0 DAE, which has no implicit constraints that relate p and q. Two dynamic degrees of freedom on $(y = y_0, z = z_0)$ are required in order to specify a unique solution. Thus determination of consistent Cauchy data requires no dynamic degrees of freedom over the initial hyperplane $(y = y_0)$, but requires a total of two side conditions on lower dimensional hyperplanes of the form $(y = y_0, z = z_0)$.

Example 6 *Consider again the simple system (3.26) presented earlier. Recall that, for Cauchy data on the hyperplane $(x_2 = 0)$, a value of either $v(x_1, 0)$ or $u(x_1, 0)$ completely determined the data. Let us consider each case in more detail. The exterior direction is x_2, so $u_{x_2} = \dot{u}$ and $v_{x_2} = \dot{v}$. The equations that must be satisfied over the hyperplane include the original equations*

$$\begin{aligned} u_{x_1} - v &= 0 \\ \dot{u} &= f_1(x_1) \end{aligned} \tag{3.49}$$

and the additional independent equation derived during index analysis

$$\dot{u}_{x_1} - \dot{v} = 0 \tag{3.50}$$

Here $r = 1$; one dynamic degree of freedom on $(x_2 = 0)$ is required to determine unique Cauchy data.

First, consider specification of v over the hyperplane, so that

$$v = h_1(x_1) \tag{3.51}$$

is appended to the system (3.49-3.50). These four equations in the four variables $\dot{u}$, u, $\dot{v}$, and v form the PDE (here a DAE) that will be used to determine the Cauchy data on $(x_2 = 0)$.

Because Cauchy data on the 1-dimensional hyperplane in $\mathbb{R}^2$ *is determined by a DAE, additional 0-dimensional Cauchy data may be required for specification of a unique solution. It is thus necessary to perform index analysis on this interior system (3.49-3.51) to determine what restrictions exist on 0-dimensional Cauchy data.*

For clarity, let $a = u$*,* $b = v$*,* $c = \dot{u} = u_{x_2}$*, and* $d = \dot{v} = v_{x_2}$*, so that the equations (3.49 - 3.51) become*

$$\begin{aligned} a_{x_1} - b &= 0 \\ c_{x_1} - d &= 0 \\ c &= f_1(x_1) \\ b &= h_1(x_1) \end{aligned} \tag{3.52}$$

Now consider a 0-dimensional subsurface $(x_2 = 0, x_1 = k_1)$ *on which additional data is to be specified. Using the standard notation for DAEs, the system is*

$$\begin{aligned} \dot{a} - b &= 0 \\ \dot{c} - d &= 0 \\ c &= f_1(x_1) \\ b &= h_1(x_1) \end{aligned} \tag{3.53}$$

Differentiation of the last three equations gives

$$\begin{aligned} \ddot{c} - \dot{d} &= 0 \\ \dot{c} &= \frac{d}{dx_1} f_1(x_1) \\ \dot{b} &= \frac{d}{dx_1} h_1(x_1) \end{aligned} \tag{3.54}$$

The second equation above may be differentiated again without producing any new variables, so also

$$\ddot{c} = \frac{d^2}{dx_1^2} f_1(x_1) \tag{3.55}$$

Two differentiations were required to derive these eight equations in the nine unknowns $\dot{a}$*,* a*,* $\dot{b}$*,* b*,* $\ddot{c}$*,* $\dot{c}$*,* c*,* $\dot{d}$*, and* d*. The index of the DAE is two, and under the assumption that* $h_1(x_1)$ *is once differentiable and* $f_1(x_1)$ *is twice differentiable with respect to* x_1*,* *one dynamic degree of freedom on* $(x_2 = 0, x_1 = k_1)$ *is required to determine uniquely consistent Cauchy data on* $(x_2 = 0)$.

The case is different if u *rather than* v *is specified. Appending*

$$u = h_2(x_1) \tag{3.56}$$

as the dynamic degree of freedom on $(x_2 = 0)$ *to the system (3.49) produces a different DAE on the hyperplane. Using the same new variables* a, b, c, *and* d, *the data must now satisfy*

$$\begin{aligned} \dot{a} - b &= 0 \\ \dot{c} - d &= 0 \\ c &= f_1(x_1) \\ a &= h_2(x_1) \end{aligned} \tag{3.57}$$

Differentiating the entire system yields

$$\begin{aligned} \ddot{a} - \dot{b} &= 0 \\ \ddot{c} - \dot{d} &= 0 \\ \dot{c} &= \frac{d}{dx_1} f_1(x_1) \\ \dot{a} &= \frac{d}{dx_1} h_2(x_1) \end{aligned} \tag{3.58}$$

Differentiating the last two equations again produces two new equations without introducing any new unknowns.

$$\begin{aligned} \ddot{c} &= \frac{d^2}{dx_1^2} f_1(x_1) \\ \ddot{a} &= \frac{d^2}{dx_1^2} h_2(x_1) \end{aligned} \tag{3.59}$$

Two differentiations were required to derive these ten equations in the ten unknowns $\ddot{a}$, $\dot{a}$, a, $\ddot{c}$, $\dot{c}$, c, $\dot{b}$, b, $\dot{d}$, *and* d. *The index of the consistent Cauchy data problem that resulted from specifying* u *rather than* v *over the hyperplane* $(x_2 = 0)$ *is again two, but in this case* $r = 0$ *and no dynamic degrees of freedom on* $(x_2 = 0, x_1 = k_1)$, *or lower-dimensional data, are required to determine unique Cauchy data on* $(x_2 = 0)$. *Here both* $f_1(x_1)$ *and* $h_2(x_1)$ *must be twice differentiable with respect to* x_1.

This result makes sense, when one considers the original system. If v is specified over $(x_2 = 0)$, the first equation in the original system (3.49) then determines u up to a constant of integration. The value of u at some point on $(x_2 = 0)$ fixes this constant of integration and fully specifies unique Cauchy data on that surface. If u is specified instead, the first equation gives v directly and no additional information is required.

Determination of consistent Cauchy data on $(x_2 = 0)$ thus requires specification of one dynamic degree of freedom on $(x_2 = 0)$. If v is specified, an additional dynamic degree of freedom on $(x_2 = 0, x_1 = k_1)$ is required to fully determine Cauchy data on $(x_2 = 0)$. If u is specified, no dynamic degrees of freedom are needed on lower dimensional hyperplanes.

Now, consider the equations that Cauchy data on the hyperplane $(x_1 = 0)$ must satisfy. Again using a dot to denote exterior derivatives, the system is

$$\begin{aligned} \dot{u} - v &= 0 \\ u_{x_2} &= f_1(x_1) \end{aligned} \tag{3.60}$$

Differentiating the second equation produces no new variables, but produces an independent equation:

$$\dot{u}_{x_2} = \frac{d}{dx_1} f_1(x_1) \tag{3.61}$$

Differentiating the first equation and the second one more time produces two new equations in two new variables, which include $\dot{v}$:

$$\begin{aligned} \ddot{u} - \dot{v} &= 0 \\ \ddot{u}_{x_2} &= \frac{d^2}{dx_1^2} f_1(x_1) \end{aligned} \tag{3.62}$$

Again under the assumption that all required derivatives exist, the index of the system with respect to x_1 is 2. There are five equations that relate the five unknowns $u, \dot{u}, \ddot{u}, v, \dot{v}$, so no dynamic degrees of freedom on $(x_1 = 0)$ may be specified arbitrarily.

The consistent Cauchy data problem is again not strictly algebraic, so lower dimensional data may be required to determine a unique solution. Let $a = u$, $b = v$, $c = \dot{u} = u_{x_1}$, $d = \ddot{u} = u_{x_1 x_1}$, and $e = \dot{v} = v_{x_1}$, and consider the hyperplane

$(x_1 = 0, x_2 = k_2)$. Using a dot to now denote differentiation in the x_2 direction, the system under consideration (3.60 - 3.62) is

$$\begin{aligned} c - b &= 0 \\ \dot{a} &= f_1(x_1) \\ d - e &= 0 \\ \dot{c} &= \frac{d}{dx_1} f_1(x_1) \\ \dot{d} &= \frac{d^2}{dx_1^2} f_1(x_1) \end{aligned} \tag{3.63}$$

Differentiating the algebraic equations produces two additional equations.

$$\begin{aligned} \dot{c} - \dot{b} &= 0 \\ \dot{d} - \dot{e} &= 0 \end{aligned} \tag{3.64}$$

Thus there are seven equations in ten unknowns, and three dynamic degrees of freedom on $(x_1 = 0, x_2 = k_2)$ are required. Five equations determine the values of $\dot{a}, \dot{b}, \dot{c}, \dot{d}$, and $\dot{e}$. Feasible specification is a, and either b or c, and either d or e.

However, note that specification of d or e is used to determine $u_{x_1x_1}$ and v_{x_1}, neither of which occur in the original equations. Three dynamic degrees of freedom on $(x_1 = 0, x_2 = k_2)$ must be specified to determine unique Cauchy for the system (3.60 - 3.62) derived during index analysis on $(x_1 = 0)$, but only two are required to determine unique Cauchy data for the original variables u, u_{x_1}, and v.

Unique Cauchy data on $(x_1 = 0)$ for the original variables requires specification of u and either u_{x_1} or v at a single point $(x_1 = 0, x_2 = k_2)$. This result again makes sense when one considers the original system. The second equation in (3.60) determines u up to a constant of integration over $(x_1 = 0)$. Equation (3.61) specifies u_{x_1} up to another constant of integration over $(x_1 = 0)$, and the first equation in (3.60) relates u_{x_1} and v on that same hyperplane. Specification of u fixes the first constant, and specification of either u_{x_1} or v fixes the second.

This does not contradict the known solution (3.32). Rather, it highlights the fact that the Cauchy data subproblems are defined only on a particular hyperplane. While u and u_{x_1} are independent over $x_1 = c_1$, they are related on $x_2 = c_2$. A single *boundary condition on u, specified over $x_2 = c_2$, may therefore provide both of the lower-dimensional specifications needed to determine unique Cauchy data on $x_1 = c_1$.*

3.6 The Navier-Stokes equations

For a larger example of this analysis, consider the two-dimensional, incompressible formulation of the Navier-Stokes equations.

$$\begin{aligned} u_t + uu_{x_1} + p_{x_1} + vu_{x_2} - \nu u_{x_1x_1} - \nu u_{x_2x_2} &= 0 \\ v_t + uv_{x_1} + vv_{x_2} + p_{x_2} - \nu v_{x_1x_1} - \nu v_{x_2x_2} &= 0 \\ u_{x_1} + v_{x_2} &= 0 \end{aligned} \tag{3.65}$$

Consider the initial hyperplane, orthogonal to t at $t = 0$. The exterior direction is along the t axis; x_1 and x_2 are interior directions. The system may be rewritten as

$$\begin{aligned} \dot{u} + uu_{x_1} + p_{x_1} + vu_{x_2} - \nu u_{x_1x_1} - \nu u_{x_2x_2} &= 0 \\ \dot{v} + uv_{x_1} + vv_{x_2} + p_{x_2} - \nu v_{x_1x_1} - \nu v_{x_2x_2} &= 0 \\ u_{x_1} + v_{x_2} &= 0 \end{aligned} \tag{3.66}$$

Differentiating the third equation with respect to the exterior direction produces another independent equation:

$$\dot{u}_{x_1} + \dot{v}_{x_2} = 0 \tag{3.67}$$

The first two equations in the original system, and the differentiated continuity equation, may be differentiated again to produce three independent equations in three new variables (which include $\dot{p}$).

$$\begin{aligned} \ddot{u} + \dot{u}u_{x_1} + u\dot{u}_{x_1} + \dot{p}_{x_1} + \dot{v}u_{x_2} + v\dot{u}_{x_2} - \nu\left(\dot{u}_{x_1x_1} + \dot{u}_{x_2x_2}\right) &= 0 \\ \ddot{v} + \dot{u}v_{x_1} + u\dot{v}_{x_1} + \dot{p}_{x_2} + \dot{v}v_{x_2} + v\dot{v}_{x_2} - \nu\left(\dot{v}_{x_1x_1} + \dot{v}_{x_2x_2}\right) &= 0 \\ \ddot{u}_{x_1} + \ddot{v}_{x_2} &= 0 \end{aligned} \tag{3.68}$$

Two differentiations with respect to t were required to uniquely determine the exterior derivatives of all variables, so the index of the Navier-Stokes equations with respect to time is 2. On the initial hyperplane, there are seven independent equations (3.66 - 3.68) that relate the eight variables $\ddot{u}, \dot{u}, u, \ddot{v}, \dot{v}, v, \dot{p}, p$, so $r = 1$ and only one dynamic degree of freedom on $(t = 0)$ exists for the specification of Cauchy data.

Typical initial conditions for the Navier-Stokes equations include specification of both u and v as dynamic degrees of freedom on $t = 0$, often $u = v = 0$ [22]. It is easy to verify that the second specification is redundant, as indicated by the index analysis. Consider the original equations (3.66) and the implicit constraint (3.67), which involve only the original variables, together with algebraic specification of u on the initial hyperplane.

$$\begin{aligned} \dot{u} + uu_{x_1} + p_{x_1} + vu_{x_2} - \nu u_{x_1x_1} - \nu u_{x_2x_2} &= 0 \\ \dot{v} + uv_{x_1} + vv_{x_2} + p_{x_2} - \nu v_{x_1x_1} - \nu v_{x_2x_2} &= 0 \\ u_{x_1} + v_{x_2} &= 0 \\ \dot{u}_{x_1} + \dot{v}_{x_2} &= 0 \\ u &= 0 \end{aligned} \tag{3.69}$$

Consistent Cauchy data, which are values of u, $\dot{u}$, v, $\dot{v}$, and p over the entire domain at $t = 0$, are a solution to this 5×5 elliptic system. The solution is uniquely determined when boundary conditions for the elliptic system are specified.

Algebraic manipulation produces the following simplified system.

$$\begin{aligned} \dot{u} + p_{x_1} &= 0 \\ \dot{v} + p_{x_2} - \nu v_{x_1x_1} &= 0 \\ v_{x_2} &= 0 \\ p_{x_1x_1} + p_{x_2x_2} &= 0 \\ u &= 0 \end{aligned} \tag{3.70}$$

This system is not strictly algebraic, so as in the previous examples, lower dimensionality degrees of freedom may be explored. Let $a = u$, $b = u_t$, $c = v$, $d = v_t$, $e = p$, and consider now the hyperplane $(t = 0, x_1 = c_1)$. The exterior direction is now x_1, so the system may be written as

$$\begin{aligned} b + \dot{e} &= 0 && (3.71) \\ d + e_{x_2} - \nu \ddot{c} &= 0 && (3.72) \\ c_{x_2} &= 0 && (3.73) \\ \ddot{e} + e_{x_2 x_2} &= 0 && (3.74) \\ a &= 0 && (3.75) \end{aligned}$$

Proceeding with determination of the index of this system with respect to x_1, differentiation of all equations save the fourth produces four additional independent equations.

$$\begin{aligned} \dot{b} + \ddot{e} &= 0 && (3.76) \\ \dot{d} + \dot{e}_{x_2} - \nu \dddot{c} &= 0 && (3.77) \\ \dot{c}_{x_2} &= 0 && (3.78) \\ \dot{a} &= 0 && (3.79) \end{aligned}$$

The third equation may be differentiated twice more without introducing any new variables, so consistent data on $(t = 0, x_1 = c_1)$ must also satisfy

$$\ddot{c}_{x_2} = 0 \qquad (3.80)$$

and

$$\dddot{c}_{x_2} = 0 \qquad (3.81)$$

Three differentiations were required to produce these 11 independent equations in the thirteen variables $a, \dot{a}, b, \dot{b}, c, \dot{c}, \ddot{c}, \dddot{c}, d, \dot{d}, e, \dot{e}, \ddot{e}$, so the index with respect to x_1 of this system (not of the original Navier-Stokes equations) is three. Two dynamic degrees of freedom are required on $(t = 0, x_1 = c_1)$.

Block decomposition of the system shows that six equations (3.75, 3.79, 3.73, 3.78, 3.80, 3.81) may be solved for the six unknowns a, $\dot{a}$, c, $\dot{c}$, $\ddot{c}$, $\dddot{c}$; two equations (3.71, 3.77) relate b, $\dot{d}$, $\dot{e}$; and three equations (3.72, 3.74, 3.76) relate $\dot{b}$, d, e, $\ddot{e}$. One specification of either b, $\dot{d}$, or $\dot{e}$, and another of either $\dot{b}$, d, e, or $\ddot{e}$, is required to determine unique Cauchy data if all specifications are to be made on a single surface orthogonal to x_1 and t. None of the variables a, $\dot{a}$, c, $\dot{c}$, $\ddot{d}$, and $\ddot{c}$ may be specified independently on $(t = 0, x_1 = c_1)$.

The first set of two equations, plus a dynamic degree of freedom assignment from the first group of three variables, is used to determine $\dot{e}$. This corresponds to a Neumann condition on pressure for Laplace's equation in (3.70). The second group is used to determine e, which corresponds to a Dirichlet condition on pressure.

It is well-known that specification of p (here e) and p_{x_1} (here $\dot{e}$) on the same line $(x_1 = c_1)$, together with Laplace's equation for p, produces an ill-posed problem [18]. Rather, either p or p_{x_1} are required on two separate hyperplanes orthogonal to the x_1 axis. Clearly, then, our index analysis provides only restrictions on allowable Cauchy data on a given surface, rather than complete information on proper boundary conditions for all problems.

Consider now the subsurface $(t = 0, x_2 = c_2)$. The exterior direction of interest is now x_2, so the system is

$$\begin{aligned} b + e_{x_1} &= 0 \\ d + \dot{e} - \nu c_{x_1 x_1} &= 0 \\ \dot{c} &= 0 \\ e_{x_1 x_1} + \ddot{e} &= 0 \\ a &= 0 \end{aligned} \tag{3.82}$$

The first, second, and last equations may be differentiated without producing any new variables, so we must also have

$$\begin{aligned} \dot{b} + \dot{e}_{x_1} &= 0 \\ \dot{d} + \ddot{e} - \nu \dot{c}_{x_1 x_1} &= 0 \\ \dot{a} &= 0 \end{aligned} \tag{3.83}$$

on $(t = 0, x_2 = c_2)$.

This is a system of eight equations in the eleven unknowns a, $\dot{a}$, b, $\dot{b}$, c, $\dot{c}$, d, $\dot{d}$, e, $\dot{e}$, $\ddot{e}$. Three equations give the values of a, $\dot{a}$, and c. Three equations relate the variables

b, $\dot{d}$, e, $\ddot{e}$, and two equations relate $\dot{b}$, c, d, $\dot{e}$. Feasible specification is thus one variable from the second group, and two from the third. Again, the first group corresponds to a Dirichlet condition on pressure. The second group includes a condition on v (here c) and a Neumann condition on pressure.

Index analysis rules out some specifications as infeasible, and in general provides only an upper bound on the number of degrees of freedom available on a particular surface. Consider planes of the form $(t = 0, x_2 = c_i)$. If v is specified over one such plane, it cannot be specified on any others. The third equation in (3.70) fixes it on all other parallel planes. Physically, this is due to the incompressibility condition and specification of $u = 0$ over the initial hypersurface. Because the fluid is incompressible and flow in the x_1 direction (u) at $t = 0$ is zero, flow in the x_2 direction (v) must be constant along lines $x_1 = constant$.

Mathematically, this appears as the third equation in the simplified original system (3.70) that says that, at $t = 0$, v does not vary with x_2. So, while index analysis indicates that three dynamic degrees of freedom are available on two parallel hyperplanes $(t = 0, x_2 = c_i)$ and $(t = 0, x_2 = c_j)$, specification of v on one takes up that degree of freedom on both.

Index analysis of the incompressible Navier-Stokes equations demonstrates that only one dependent variable may be independently specified over $(t = 0)$. If that specification is $u = 0$, added information on 1-dimensional hyperplanes within $(t = 0)$ is required to determine unique, consistent Cauchy data on $(t = 0)$. On hyperplanes of the form $(t = 0, x_1 = c_1)$, the only allowable dynamic degrees of freedom are combinations of p and p_{x_1}, while on hyperplanes of the form $(t = 0, x_2 = c_2)$, the only allowable dynamic degrees of freedom are combinations of p, p_{x_2}, and v. Depending on how these degrees of freedom are specified, additional data on 0-dimensional hyperplanes may be required to complete determination of unique Cauchy data on $(t = 0)$. The example also demonstrates that, while index analysis can provide useful information about allowable boundary conditions for an elliptic Cauchy data problem, it does not provide all the information needed to form a well-posed problem.

3.7 Relating the differentiation and algebraic indices

The differentiation index with respect to t and the algebraic t-index for linear systems are equivalent.

Theorem 3.7.1 *The differentiation index with respect to t and the algebraic t-index of a linear system of first order in t are equal.*

Proof. Let the differentiation index with respect to t be ν_t^D. Then the smallest smoothly 1-full derivative array is $\mathcal{A}_{\nu_t^D+1}$, so $\mathbf{u}_t$ depends explicitly on up to ν_t^D derivatives with respect to t of an arbitrary forcing function $\mathbf{f}(x,t)$. Therefore $\mathbf{u}$ depends explicitly on up to $\nu_t^D - 1$ partial derivatives with respect to t of $\mathbf{f}$, and $\mathbf{R}(s,z)$ must contain at least one rational function in s with highest powers $\frac{s^{(j+\nu_t^D-1)}}{s^j}$, so the algebraic t-index ν_t^A must be $\nu_t^A = (\nu_t^D - 1) + 1 = \nu_t^D$. □

As noted earlier, the differentiation index captures only *explicit* dependence of the solution on derivatives of the forcing functions. For a linear system for which the algebraic index may be defined, the algebraic index will also have the same property. The question of smoothness requirements on forcing functions will be taken up in the next chapter.

3.8 Higher order systems

Any higher order equation may be written as an equivalent system of first order equations [40]. However, here "equivalent" means only that the solution in the original variables is identical to that of the higher order system. It does not mean that the index of the first order system is equal to the index of the original system.

Consider first the index with respect to x_j of a system that is first order in x_j, but higher order in $x_{i\neq j}$. Index analysis of linear and linear x_j-varying systems based on the derivative array equations applies directly to such systems, because the interior partial differential operators included in P_{Ij} may be of any order. Reducing the order of such a system is therefore not necessary to determine the index with respect to x_j, and may in fact increase the index, as shown by the following simple example.

Example 7 *Consider the heat equation.*

$$u_t - u_{xx} = 0 \tag{3.84}$$

The index of this system with respect to t is 0, while the index with respect to t of the first order form of the heat equation

$$\begin{aligned} u_t - v_x &= 0 \\ u_x &= v \end{aligned} \tag{3.85}$$

is 1.

Consider next a general system of m equations, some of which are second order in the variable of interest t.

$$\mathbf{F}(\mathbf{u}_{tt}, \mathbf{u}_t, \mathbf{u}_x, \mathbf{u}, t, x) = \mathbf{0} \tag{3.86}$$

Divide the dependent variables into two groups $\mathbf{w}$ and $\mathbf{y}$, where $\mathbf{w}$ consists of all dependent variables for which a second partial derivative with respect to t appears in the system, and $\mathbf{y}$ consists of all other dependent variables. Let the dimension of $\mathbf{w}$ be p and of $\mathbf{y}$ be q, so that $p + q = m$. Written in terms of these new variables, the system is

$$\mathbf{F}(\mathbf{w}_{tt}, \mathbf{w}_t, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) = \mathbf{0} \tag{3.87}$$

Reducing the order in t by introducing new variables and equations produces the following system.

$$\begin{aligned} \mathbf{w}_t &= \mathbf{v} \\ \mathbf{F}(\mathbf{v}_t, \mathbf{v}, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) &= \mathbf{0} \end{aligned} \tag{3.88}$$

Let the index of the first system (3.87) with respect to t be defined as the minimum number of differentiations with respect to t required in order to determine $\mathbf{y}_t$ as a continuous function of $\mathbf{y}$, $\mathbf{w}$, t, and x, and $\mathbf{w}_{tt}$ as a continuous function of $\mathbf{w}_t$, $\mathbf{w}$, $\mathbf{y}$, t, and x. The index of the reduced order system (3.88) with respect to t is already well-defined.

Theorem 3.8.1 *The index of a second order system in t with respect to t and of the equivalent first order system in t are equal.*

Proof. Compare the derivative array equations in t for the second order system

$$\begin{aligned} \mathbf{F}(\mathbf{w}_{tt}, \mathbf{w}_t, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) &= \mathbf{0} \\ \frac{\partial}{\partial t}\mathbf{F}(\mathbf{w}_{tt}, \mathbf{w}_t, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) &= \mathbf{0} \\ \left(\frac{\partial}{\partial t}\right)^2 \mathbf{F}(\mathbf{w}_{tt}, \mathbf{w}_t, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) &= \mathbf{0} \\ &\vdots \end{aligned} \tag{3.89}$$

to the derivative array equations for the equivalent first order system formed by differentiating $\mathbf{F}$ only.

$$\begin{aligned} \begin{bmatrix} \mathbf{w}_t - \mathbf{v} \\ \mathbf{F}(\mathbf{v}_t, \mathbf{v}, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) \end{bmatrix} &= \mathbf{0} \\ \frac{\partial}{\partial t}\mathbf{F}(\mathbf{v}_t, \mathbf{v}, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) &= \mathbf{0} \\ \left(\frac{\partial}{\partial t}\right)^2 \mathbf{F}(\mathbf{v}_t, \mathbf{v}, \mathbf{w}_x, \mathbf{w}, \mathbf{y}_t, \mathbf{y}_x, \mathbf{y}, t, x) &= \mathbf{0} \\ &\vdots \end{aligned} \tag{3.90}$$

Clearly the derivative arrays in t for the two systems are the equivalent; the only difference is the change of variables $\mathbf{w}_{tt} = \mathbf{v}_t$ and $\mathbf{w}_t = \mathbf{v}$ and the augmented first row. If k is the smallest integer such that the first $k+1$ rows of the derivative array for the second order system determine $\mathbf{y}_t$ as a continuous function of $\mathbf{y}$, $\mathbf{w}$, t, and x, and $\mathbf{w}_{tt}$ as a continuous function of $\mathbf{w}_t$, $\mathbf{w}$, $\mathbf{y}$, t, and x, it must also be the smallest integer such that the first $k+1$ rows of the derivative array for the first order system determine $\mathbf{y}_t$ as a continuous function of $\mathbf{y}$, $\mathbf{w}$, t, and x, and $\mathbf{v}_t$ as a continuous function of $\mathbf{v}$, $\mathbf{w}$, $\mathbf{y}$, t, and x. $\mathbf{w}_t$ is given as a continuous function of $\mathbf{v}$ for $k = 0$. The index of both systems with respect to t is therefore k. □

So, reducing the order in x_i is unnecessary for determination of, and may in fact alter, the index with respect to $x_{j \neq i}$. Reducing the order in x_j will not alter the index with respect to x_j.

Chapter 4

Generalized Characteristic Analysis

4.1 Introduction

The differentiation index analysis presented in the previous chapter provides valuable information about distributed unit models. In particular, it gives the number of dynamic degrees of freedom, as well as providing insight into what specifications of initial data are consistent with the equations. However, it does not directly address the question of whether or not a solution exists, is unique, or depends continuously on its data. These issues of well-posedness are unique to distributed unit models. This chapter will present methods of analysis that address these questions, based on a generalization of classical characteristic analysis of hyperbolic systems to more general nonhyperbolic models.

Before talking about the existence and uniqueness of a solution, and the dependence of the solution on its data, it is necessary to specify more precisely what is meant by the term *solution.* The analysis in this chapter assumes that the term solution refers to **strong solutions**. A strong solution to a partial differential system is a function that satisfies the governing equations pointwise everywhere. For a first order system, this implies C^1 continuity in all directions. Initial and boundary data is typically used either to build the functional form of the solution, or to restrict the space of basis functions from which the solution is constructed. A solution built in the former manner will be called an **unrestricted solution**; if data is used to restrict the function space from which the solution is drawn, that solution will be called a **restricted solution**.

Linear, first order systems over two independent variables t and x will be considered. Higher order systems may always be transformed to a larger but equivalent first order system through introduction of new variables for higher order terms [40]. All required Fourier and Laplace transforms are assumed to be well defined. Unless otherwise stated, the term *index* refers to the differentiation index defined in the previous chapter.

4.2 Systems with simple forcing

Consider a linear, two-dimensional system with simple forcing

$$\mathbf{A}\mathbf{u}_t + \mathbf{B}\mathbf{u}_x = \mathbf{f}(t, x) \tag{4.1}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, $\mathbf{u} \in \mathbb{R}^n$, and $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^n$. A system of this form is regular [12] iff the coefficient matrices form a regular pencil; that is, there exist some real constants s, z such that $|s\mathbf{A} + z\mathbf{B}| \neq 0$.

Under the assumption that the coefficient matrices form a regular pencil, every linear system with simple forcing is equivalent to one of the following form, which will be referred to as both its **canonical form** and its **generalized characteristic form**.

$$\begin{bmatrix} \mathbf{J} & & \\ & \mathbf{N}_1 & \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}_t + \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{I} & \\ & & \mathbf{N}_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}_x = \begin{bmatrix} \mathbf{f}_1(t, x) \\ \mathbf{f}_2(t, x) \\ \mathbf{f}_3(t, x) \end{bmatrix} \tag{4.2}$$

Here $\mathbf{J}$ is an invertible lower Jordan matrix, and $\mathbf{N}_1$ and $\mathbf{N}_2$ are lower Jordan nilpotent matrices. The first block row will be called the **hyperbolic part**, the second the **parabolic part**, and the third the **differential part**, of dimension n_h, n_p, and n_d respectively. Let ν_i be the nilpotency of $\mathbf{N}_i$. The canonical form is constructed in the same manner as the canonical form of a DAE, with the generalized eigenvalues ordered to produce the three desired block rows. Finally, let the **degeneracy** of a Jordan block be defined as one less than the dimension of the block; let the degeneracy of the system be defined as the maximum degeneracy of any Jordan block.

Theorem 4.2.1 *The differentiation index with respect to t, ν_t, of a linear system with simple forcing is equal to ν_1.*

Proof. The hyperbolic and differential parts of the system give $\mathbf{v}_{1_t}$ and $\mathbf{v}_{3_t}$ as continuous functions of $\mathbf{v}_x$, t, and x. The smallest derivative array with respect to t [54] for the parabolic part that is 1-full has $\nu_1 + 1$ block rows, so the index of the system with respect to t is ν_1. □

Corollary 4.2.2 *The differentiation index with respect to x, ν_x, of a linear system with simple forcing is equal to ν_2.*

Remark 1 *A linear PDE with simple forcing may have arbitrary index with respect to any coordinate direction.*

Remark 2 *All systems with a parabolic part have $\nu_t \geq 1$, and all systems with a differential part have $\nu_x \geq 1$.*

Remark 3 *Only systems that consist strictly of a hyperbolic part may have both indices equal to zero.*

The differentiation index of a linear DAE provides an upper bound on the order of derivatives of the forcing functions that appear in the solution. This is not true for PDEs. Because PDEs may be coupled through derivative terms, forcing functions that appear in the solution for one dependent variable may appear via a partial derivative of that variable in the solution for another dependent variable. Consider a single block of the hyperbolic subsystem of a linear PDE with simple forcing. Substituting

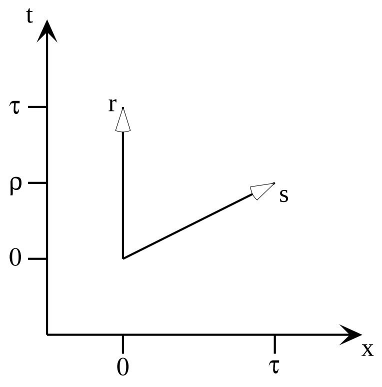

Figure 4-1: Unit r and s vectors mapped into the (t, x) plane

for the subdiagonal partial derivatives with respect to t gives a system of the form

$$\begin{aligned}
k\bar{v}_{1t} &= -\bar{v}_{1x} + f_1 \\
k\bar{v}_{2t} &= -\bar{v}_{2x} + f_2 - (-\bar{v}_{1x} + f_1) \\
k\bar{v}_{3t} &= -\bar{v}_{3x} + f_3 - (-\bar{v}_{2x} + f_2 - (-\bar{v}_{1x} + f_1)) \\
&\vdots
\end{aligned} \tag{4.3}$$

where $k \in \mathbb{R}$, $k \neq 0$, and $\bar{\mathbf{v}} \subset \mathbf{v}_1$.

Consider the Cauchy problem for this block on an infinite domain in x, with analytic initial data $\bar{\mathbf{v}}(0, x)$. The smoothness of $\bar{v}_1(t, x)$ is then one greater than the smoothness of $f_1(t, x)$. The smoothness of $\bar{v}_2$ depends not only on the smoothness of $f_2(t, x)$, but also on the smoothness of $\bar{v}_{1x}(t, x)$, which is one greater than the smoothness of $f_{1x}(t, x)$. For a degenerate hyperbolic block of dimension 3, $\bar{\mathbf{v}}(t, x) \in C^1$ requires that $f_1(t, x)$ be C^2, $f_2(t, x)$ be C^1, and $f_3(t, x)$ be C^0 in x. The canonical form could instead be defined with the $\mathbf{J}$ block of the hyperbolic part appearing in the second coefficient matrix; the corresponding Cauchy problem in t would show analogous explicit smoothness requirements on $\mathbf{f}$ in t.

This implicit dependence of the solution on derivatives of the forcing functions is made explicit by a change to a different coordinate system. Let the transformation from (t, x)-space to (r, s)-space for a single Jordan block $\mathbf{J}_i$ of the hyperbolic part be given by

$$\begin{bmatrix} \tau_i & \rho_i \\ 0 & \tau_i \end{bmatrix} \begin{bmatrix} r_i \\ s_i \end{bmatrix} = \begin{bmatrix} t \\ x \end{bmatrix} \tag{4.4}$$

where ρ_i is the value on the diagonal of $\mathbf{J}_i$ and $\tau_i = 1$. In these independent variables, the equations have the same general form shared by blocks in the differential and parabolic subsystems, which is simply

$$\mathbf{N}\bar{\mathbf{v}}_a + \mathbf{I}\bar{\mathbf{v}}_b = \bar{\mathbf{f}}(a, b) \tag{4.5}$$

Here $\bar{\mathbf{f}} : \mathbb{R}^2 \to \mathbb{R}^m$, and $\mathbf{N} \in \mathbb{R}^{m\times m}$ is a matrix of nilpotency m, with unity on the first subdiagonal and zeros everywhere else. Also note that $\bar{\mathbf{f}} \subset \mathbf{f}_j$ and $\bar{\mathbf{v}} \subset \mathbf{v}_j$, $j \in \{1, 2, 3\}$.

The solution to a system of this general form is built recursively, and is a polynomial in b with coefficients that, in general, may be functions of a. Integrating each equation with respect to b and substituting the result into the subsequent equation yields

$$\begin{aligned}
\bar{v}_1 &= c_1(a) + \int \bar{f}_1 db \\
\bar{v}_2 &= -bc_1'(a) + c_2(a) + \int \left[\bar{f}_2 - \int \bar{f}_{1_a} db\right] db \\
\bar{v}_3 &= \frac{b^2}{2}c_1''(a) - bc_2'(a) + c_3(a) + \int \left[\bar{f}_3 - \int \left[\bar{f}_{2_a} - \int \bar{f}_{1_{aa}} db\right] db\right] db \\
&\vdots
\end{aligned} \tag{4.6}$$

This representation makes explicit the dependence of the smoothness of $\bar{\mathbf{v}}$ on the data $\mathbf{c}$ and forcing functions $\bar{\mathbf{f}}$. For example, the degree of smoothness of $\bar{v}_1$ in the b direction is one greater than the degree of smoothness of $\bar{f}_1$ in b, while the degree of smoothness of $\bar{v}_1$ in a is equal to the lesser of the degree of smoothness in a of the data c_1 and forcing function $\bar{f}_1$. For $\bar{v}_m$ to be C^1 in a, the forcing function $\bar{f}_i$ must be at least C^{m-i+1} in a, and the data c_i must be at least C^{m-i+1}. For $\bar{v}_m$ to be C^1 in b, $\bar{f}_{i\leq m}$ must be continuous.

Putting a block of the hyperbolic subsystem into this form (4.5) required a coordinate change. Because $a = r$ and

$$\frac{\partial}{\partial r} = \frac{1}{\tau}\frac{\partial}{\partial t} - \frac{\rho}{\tau^2}\frac{\partial}{\partial x}$$

$\bar{f}_i$ must be at least C^{m-i+1} in both x and t in order for $\bar{v}_m$ to be C^1 in r.

Theorem 4.2.3 *The maximum order of derivatives of forcing functions and data that appear in the solution of a two-dimensional, linear PDE with simple forcing is equal to ν, the degeneracy of the coefficient matrix pair.*

Proof. The solution for a Jordan block of dimension m depends on up to $m-1$ exterior partial derivatives of the data and forcing functions, and by definition $\max(m_i) = \nu + 1$. $\square$

This result shows that it is the *degeneracy*, rather than the index, that gives sufficient conditions for the forcing function and data smoothness required for existence of a continuous or a smooth solution. A system that consists strictly of a hyperbolic subsystem will have index 0 with respect to both t and x, yet derivatives of the forcing functions may appear in the solution. Because the forcing terms $\mathbf{f}_i$ in the canonical form are linear combinations of the original forcing functions $\mathbf{f}$, if every element of $\mathbf{f}$ is ν-times differentiable with respect to both t and x, then every $\bar{f}_j$ will possess all partial derivatives required for existence of a continuous solution. Similarly, if all arbitrarily specified data is ν-times differentiable, then all required derivatives of data will exist. Increasing these sufficient differentiability requirements by one guarantees a smooth solution.

A system of this generic form (4.5) is equivalent to an ODE in b. Applying the partial differential operator $(\mathbf{N}D_a + \mathbf{I}D_b)^* = \sum_{i=0}^{m-1}(-1)^i \mathbf{N}^i D_a^i D_b^{(m-i)}$, where $D_a = \frac{\partial}{\partial a}$ and $D_b = \frac{\partial}{\partial b}$, to the system produces

$$(\mathbf{N}D_a + \mathbf{I}D_b)^*(\mathbf{N}\bar{\mathbf{v}}_a + \mathbf{I}\bar{\mathbf{v}}_b) = \mathbf{I}D_b^{m+1}\bar{\mathbf{v}} = (\mathbf{N}D_a + \mathbf{I}D_b)^*\bar{\mathbf{f}}(a,b) \tag{4.7}$$

Because it is equivalent to an interior partial differential system along lines of constant a, a block of this form may be viewed as a generalization of the characteristic form of a one-way wave. The exterior partial derivatives are governed entirely by the data and the forcing functions. Furthermore, the degeneracy of the wave means that one or more exterior partial derivatives of a particular dependent variable must exist in order for subsequent dependent variables to exist.

The canonical form may thus be viewed as a generalization of the characteristic form of a hyperbolic system. Because each Jordan block is equivalent to a fully determined ODE along a particular direction b in the (t, x) plane, it provides a set of compatibility conditions that restrict Cauchy data on surfaces of the form $a = constant$. For example, Cauchy data for the parabolic part on $t = k_1$ is uniquely determined by data specified on some point $(t = k_1, x_0)$, and the forcing function $\mathbf{f}_2$. Therefore, the number of dynamic degrees of freedom that may be arbitrarily specified on $t = 0$ is equal to $n_h + n_d$.

Under the assumption that the problem will be solved as an evolution problem in t, data cannot be specified at a later time and used to determine a solution at an earlier time. Data for the hyperbolic blocks consist of arbitrary functions of r, which is along either the t or x coordinate directions depending on the coordinate transformation employed, must be prescribed on $(x = x_1)$ for hyperbolic blocks with $\tau_i/\rho_i > 0$, or on $(x = x_2)$ for blocks with $\tau_i/\rho_i < 0$, and on $t = t_0$ in either case. Data for the differential blocks consist of arbitrary functions of x, which must be specified on the initial line $x = x_0$. Because there is no righthand side dependence on $\mathbf{u}$, data may be specified on $(x = x_1)$ or $(x = x_2)$ for the parabolic part.

The dependence of solutions to first order linear systems of the form

$$\mathbf{u}_t + \mathbf{B}\mathbf{u}_x = \mathbf{f}(t, x) \tag{4.8}$$

on their initial data is well-studied [44, 81]. If $\mathbf{B}$ is not diagonalizable, but all eigenvalues of $\mathbf{B}$ are real, the solution involves derivatives of the data and is weakly well-posed. If any characteristic direction contains a nonzero imaginary component, the system will show exponential dependence on the frequency of perturbations to data and will thus be ill-posed, regardless of the degeneracy of that characteristic.

Systems of the form under consideration here (4.1) do not necessarily have the form of a hyperbolic or weakly hyperbolic system (4.8). As formulated here, the "initial data" for blocks of both the parabolic and hyperbolic subsystems are really boundary conditions (arbitrary functions of t). Note that one may reformulate the canonical form by moving $\mathbf{J}$ to the second coefficient matrix and inverting its diagonal entries; in this case the coordinate system in which a block is an interior PDE has r_i parallel to the x axis. Such a reformulation demonstrates that the dependence of the unrestricted solution of the hyperbolic subsystem on its initial data is the same as on its boundary data.

The dependence of the unrestricted solution on its data is governed by the generalized eigenvalues of the coefficient matrix pair, in an analogous manner to the case of a weakly hyperbolic system. This will be shown in the following theorem. The proof[1] employs the analytical solution [10] of the DAE that results from Fourier transforms in either t or x.

Lemma 4.2.4 *The unrestricted solution to a regular, first order system with simple forcing depends continuously on its initial data iff the differential and hyperbolic parts of the coefficient matrix pencil are of degeneracy zero with strictly real eigenvalues.*

Proof. Because the coefficient matrices form a regular pencil, the homogeneous system is equivalent to one which, in Fourier space, has the form

$$\begin{bmatrix} \mathbf{I} & \\ & \mathbf{N} \end{bmatrix} \hat{\mathbf{v}}_t + i\omega \begin{bmatrix} \mathbf{J} & \\ & \mathbf{I} \end{bmatrix} \hat{\mathbf{v}} = \mathbf{0}$$

$$\tilde{\mathbf{A}}\hat{\mathbf{v}}_t + \tilde{\mathbf{B}}\hat{\mathbf{v}} = \mathbf{0}$$

The solution to this DAE, as described in section 2.3.7, is given by

$$\hat{\mathbf{v}}(t,\omega) = e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}\tilde{\mathbf{A}}\tilde{\mathbf{A}}^D\hat{\mathbf{v}}(0,\omega)$$

where

$$\tilde{\mathbf{A}}^D\tilde{\mathbf{B}} = \begin{bmatrix} i\omega\mathbf{J} & \\ & \mathbf{0} \end{bmatrix} \qquad \tilde{\mathbf{A}}\tilde{\mathbf{A}}^D = \begin{bmatrix} \mathbf{I} & \\ & \mathbf{0} \end{bmatrix}$$

Taking the norm of both sides gives

$$||\hat{\mathbf{v}}(t,\omega)|| = ||e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}\tilde{\mathbf{A}}\tilde{\mathbf{A}}^D\hat{\mathbf{v}}(0,\omega)||$$

[1] An alternate method of proof that considers each of the (decoupled) parts of the canonical form separately could also have been employed.

By the definition of the norm,

$$||e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}\tilde{\mathbf{A}}\tilde{\mathbf{A}}^D\hat{\mathbf{v}}(0,\omega)|| \leq ||e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}||\ ||\tilde{\mathbf{A}}\tilde{\mathbf{A}}^D||\ ||\hat{\mathbf{v}}(0,\omega)||$$

Finally, note that $||\tilde{\mathbf{A}}\tilde{\mathbf{A}}^D|| = 1$, so

$$||e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}||\ ||\tilde{\mathbf{A}}\tilde{\mathbf{A}}^D||\ ||\hat{\mathbf{v}}(0,\omega)|| = ||e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}||\ ||\hat{\mathbf{v}}(0,\omega)||$$

First assume that all eigenvalues are strictly real and nondegenerate. Then, $\mathbf{J} = \mathbf{\Lambda}$ with $\Lambda_{jj} \in \mathbb{R}$ so $||e^{-i\omega\mathbf{\Lambda}t}|| = 1$ and there exists a C_t independent of ω such that

$$||e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}||\ ||\hat{\mathbf{v}}(0,\omega)|| \leq C_t||\hat{\mathbf{v}}(0,\omega)||$$

Gathering all of these inequalities together,

$$||\mathbf{v}(t,\omega)|| \leq C_t||\hat{\mathbf{v}}(0,\omega)||$$

By Parseval's equation the result holds in (t,x) space as well, and by Duhamel's principle the result holds for simple forcing.

For the converse, assume that the system depends continuously on its initial data. Then there exists a finite C_t independent of ω such that

$$||\mathbf{v}(t,\omega)|| \leq C_t||\hat{\mathbf{v}}(0,\omega)||$$

so

$$C_t \geq ||e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}||$$

for all $\omega \in \mathbb{R}$.

Recall that the magnitude of a complex number $a+bi$, with $a,b \in \mathbb{R}$, is given by

$$|a+bi| = \sqrt{a^2+b^2} \tag{4.9}$$

and also Euler's formula

$$e^{bi} = \cos(b) + i\sin(b) \tag{4.10}$$

Finally, recall that the magnitude of a matrix of dimension n is bounded from below by the maximum magnitude of a single element.

$$||\mathbf{A}|| \geq \max_{i,j} |A_{ij}| \tag{4.11}$$

Now, suppose that there is an eigenvalue λ that corresponds to a Jordan block $\mathbf{J}$ of dimension 2 or greater, and let λ be strictly real. The exponential matrix $e^{-\tilde{\mathbf{A}}^D\tilde{\mathbf{B}}t}$ has a block of the form

$$e^{-i\omega t\mathbf{J}} = \begin{bmatrix} e^{-i\omega t\lambda} & & & & \\ (-i\omega t\lambda)e^{-i\omega t\lambda} & e^{-i\omega t\lambda} & & & \\ \frac{1}{2}(-i\omega t\lambda)^2 e^{-i\omega t\lambda} & (-i\omega t\lambda)e^{-i\omega t\lambda} & e^{-i\omega t\lambda} & & \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{4.12}$$

Each term in the matrix exponential block has the form

$$c_{ij}(-i\omega t\lambda)^{k_{ij}} e^{-i\omega t\lambda} \tag{4.13}$$

so, assuming that the use of i as the imaginary number or an index of summation is clear from context,

$$||e^{-i\omega t\mathbf{J}}|| \leq n \max_{i,j} |e_{ij}^{-i\omega t\mathbf{J}}| = n \max_{i,j} |c_{ij}(-i\omega t\lambda)^{k_{ij}} e^{-i\omega t\lambda}| \tag{4.14}$$

Note that, by Euler's formula and the definition of the magnitude of a complex number,

$$\begin{aligned} |c_{ij}(-i\omega t\lambda)^{k_{ij}} e^{-i\omega t\lambda}| &= |c_{ij}(-i\omega t\lambda)^{k_{ij}}(\cos(-\omega t\lambda) + i\sin(-\omega t\lambda))| \\ &= |c_{ij}(-\omega t\lambda)^{k_{ij}}| \\ &= |c_{ij}(\omega t\lambda)^{k_{ij}}| \end{aligned} \tag{4.15}$$

Now, let i and j be the indices that maximize the above expression (4.14); then let $c = n|c_{ij}(\lambda)^{k_{ij}}|$, $k = k_{ij}$, and note that $t > 0$. Clearly

$$|c_{ij}(\omega t\lambda)^{k_{ij}}| = c|\omega|^k t^k \tag{4.16}$$

and therefore

$$||e^{-i\omega t\mathbf{J}}|| \geq c|\omega|^k t^k \tag{4.17}$$

for some $c \in \mathbb{R}$, in contradiction to the bound independent of ω.

Similarly, if there is an eigenvalue $\lambda = a + bi$ with a nonzero complex part $b \neq 0$, then

$$||e^{-\tilde{\mathbf{A}}^D \tilde{\mathbf{B}} t}|| \geq |e^{-i\omega\lambda t}| = |e^{-i\omega a t} e^{\omega b t}|$$

in contradiction to the bound independent of ω. □

Taking the Fourier transform in t rather than x gives the analogous result for the parabolic part.

Corollary 4.2.5 *The unrestricted solution to a regular, first order system with simple forcing depends continuously on its boundary data iff the hyperbolic and parabolic parts of the coefficient matrix pencil are of degeneracy zero with strictly real eigenvalues.*

Taken together, this lemma and its corollary give the desired result.

Theorem 4.2.6 *The unrestricted solution to a regular, first order system with simple forcing depends continuously on its data iff the coefficient matrix pencil is of degeneracy zero with strictly real eigenvalues.*

Note that systems of nonzero degeneracy but with strictly real eigenvalues are ill-posed in the strict sense but are considered weakly well-posed, given proper initial and boundary data. For linear systems with simple forcing, weakly well-posed and strongly ill-posed problems may be easily distinguished from each other; if any generalized eigenvalue of the coefficient matrix pair has a nonzero imaginary part, the system is strongly ill-posed, while if all generalized eigenvalues are strictly real but one or more has nonzero degeneracy, the system is weakly well-posed.

4.3 Systems with linear forcing

Introduction of forcing functions that include a linear function of the dependent variables slightly complicates the analyses of the previous section. Such a system has the form

$$\mathbf{A}\mathbf{u}_t + \mathbf{B}\mathbf{u}_x = \mathbf{f}(t, x) - \mathbf{C}\mathbf{u} \tag{4.18}$$

with $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{n\times n}$, $\mathbf{u} \in \mathbb{R}^n$, and $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^n$. Transforming the coefficient matrices to their Weierstrass canonical form does not in general produce decoupled subsystems, so the index results of the previous section no longer hold. Also, singularity of the coefficient matrix pencil no longer precludes well-posedness; here the system is regular iff $\exists s, z \in \mathbb{R}$ such that $|s\mathbf{A} + z\mathbf{B} + \mathbf{C}| \neq 0$ [12]. Finally, introduction of the lower order term $\mathbf{Cu}$ may make weakly well-posed systems strongly ill-posed.

Under the assumption that $(\mathbf{A}, \mathbf{B})$ form a regular pencil, the canonical form of the system is

$$\begin{bmatrix} \mathbf{J} & & \\ & \mathbf{N}_1 & \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}_t + \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{I} & \\ & & \mathbf{N}_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}_x + \mathbf{C}^*\mathbf{u} = \begin{bmatrix} \mathbf{f}_1(t,x) \\ \mathbf{f}_2(t,x) \\ \mathbf{f}_3(t,x) \end{bmatrix} \tag{4.19}$$

where again $\mathbf{J}$ is a lower Jordan matrix, and $\mathbf{N}_1$ and $\mathbf{N}_2$ are lower Jordan matrices of nilpotencies ν_1 and ν_2 respectively.

Although it is in general not true, if $\mathbf{C}^*$ is lower triangular, one can still get the index by inspection of the canonical form. A system in canonical form for which $\mathbf{C}^*$ lower triangular is in some sense analogous to a DAE in Hessenberg form [8]. In particular, theorem 4.2.1 and corollary 4.2.2 still hold, and may be restated as follows.

Theorem 4.3.1 *The differentiation index with respect to t, ν_t, of a linear system in canonical form, with linear forcing and $\mathbf{C}^*$ lower triangular, is equal to ν_1.*

Proof. The hyperbolic and differential parts of the system give $\mathbf{v}_{1_t}$ and $\mathbf{v}_{3_t}$ as continuous functions of $\mathbf{v}_x$, t, and x. The smallest derivative array with respect to t [54] for the parabolic part that is 1-full has $\nu_1 + 1$ block rows, so the index of the system with respect to t is ν_1. □

Corollary 4.3.2 *The differentiation index with respect to x, ν_x, of a linear system in canonical form, with linear forcing and $\mathbf{C}^*$ lower triangular, is equal to ν_2.*

Assuming a regular system, decoupled subsystems may be obtained by handling only the interior partial derivatives in the Laplace domain. Taking the system to the

Laplace domain for x gives

$$\mathbf{A}\tilde{\mathbf{u}}_t + \mathbf{D}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}(t, z) \tag{4.20}$$

where $\tilde{\mathbf{u}} = \mathcal{L}(\mathbf{u}, x)$, $\tilde{\mathbf{f}} = \mathcal{L}(\mathbf{f}, x)$, and $\mathbf{D} = (z\mathbf{B} + \mathbf{C})$. Because P, the set of all rational functions in the complex variable z, forms a field over standard addition and multiplication, there exist square, invertible matrices $\mathbf{P}, \mathbf{Q} \in P^{n \times n}$ that take $\mathbf{D}, \mathbf{A} \in P^{n \times n}$ to their Weierstrass canonical form.

Multiplying the system on the left by this $\mathbf{P}$ and introducing new variables $\tilde{\mathbf{v}}(t, z) = \mathbf{Q}^{-1}\tilde{\mathbf{u}}(t, z)$ produces a canonical form that again consists of two decoupled subsystems.

$$\begin{bmatrix} \mathbf{I} & \\ & \mathbf{N} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}}_1 \\ \tilde{\mathbf{v}}_2 \end{bmatrix}_t + \begin{bmatrix} \mathbf{J} & \\ & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}}_1 \\ \tilde{\mathbf{v}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{g}}_1(t, z) \\ \tilde{\mathbf{g}}_2(t, z) \end{bmatrix} \tag{4.21}$$

$\mathbf{J}$ is a lower Jordan matrix, and $\mathbf{N}$ is a lower Jordan matrix of nilpotency ν. The index of this system with respect to t is ν, and $r = dim(\tilde{\mathbf{v}}_1)$ dynamic degrees of freedom must be specified on $t = 0$ in order to determine a unique trajectory in t.

The first block row is a fully determined differential system with respect to t. In analogy with the analysis of a DAE, call this the t-differential part. The second block row may be solved directly. Let it be called the t-algebraic part. The solution to the t-algebraic part, given by

$$\tilde{\mathbf{v}}_2 = \sum_{i=0}^{\nu-1} (-1)^i \mathbf{N}^i \left(\frac{\partial}{\partial t}\right)^i \tilde{\mathbf{g}}_2(t, z) \tag{4.22}$$

depends on up to $\nu - 1$ partial derivatives of the forcing functions with respect to t. No data is needed over surfaces of the form $t = constant$ to determine a unique solution of the t-algebraic part, although data may be required on surfaces of lower dimensionality upon transformation back from the Laplace domain [54].

Example 8 *Consider the following system.*

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_t + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_x + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{4.23}$$

Taking x to the Laplace domain, multiplication on the left by

$$\mathbf{P} = \begin{bmatrix} -z & 1 \\ 1 & 0 \end{bmatrix}$$

and introduction of new variables

$$\hat{\mathbf{v}} = \begin{bmatrix} 1 & 0 \\ z & 1 \end{bmatrix} \hat{\mathbf{u}}$$

produces

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \end{bmatrix}_t + \begin{bmatrix} -z^2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \end{bmatrix} = \begin{bmatrix} \hat{f}_2 - z\hat{f}_1 \\ \hat{f}_1 \end{bmatrix} \tag{4.24}$$

Here $\mathbf{J} = [-z^2]$ and $\mathbf{N} = [0]$. The nilpotency of $\mathbf{N}$ is one, so the index of this system with respect to t is one. There is one dynamic degree of freedom on $t = 0$.

One might be tempted to view the polynomial-valued coefficient matrices of the system in the Laplace domain as parameterized coefficient matrices, and ask how specific numerical values of z and/or s might alter the canonical form. That view does not apply here, however. The polynomials in the coefficient matrices represent operators; they are not functions to be evaluated. All arithmetic operations such as calculation of the canonical form are performed on the polynomials themselves, rather than on the result of evaluating the polynomials at specific numeric values of s or z.

A characteristic interpretation of the system after partial transformation to the Laplace domain does not provide the same information on boundary condition requirements that it does in the (t, x) domain. In the linear forcing case, then, the characteristic and index analyses must diverge.

Consider again our original system with linear forcing (4.18). If the coefficient matrices form a regular pencil, the canonical form of the system is

$$\begin{bmatrix} \mathbf{J} & & \\ & \mathbf{N}_1 & \\ & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}_t + \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{I} & \\ & & \mathbf{N}_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}_x = \begin{bmatrix} \mathbf{g}_1(\mathbf{v}) + \mathbf{f}_1(t,x) \\ \mathbf{g}_2(\mathbf{v}) + \mathbf{f}_2(t,x) \\ \mathbf{g}_3(\mathbf{v}) + \mathbf{f}_3(t,x) \end{bmatrix} \tag{4.25}$$

Here the block rows may be coupled through the forcing functions.

Lemma 4.3.3 *For a linear, first order system over two independent variables t and x, if $\nu_t = 0$ or $\nu_x = 0$, then the coefficient matrices form a regular pencil.*

Proof. If $\nu_t = 0$, then by definition the Jacobian of the system with respect to $\mathbf{u}_t$, or $J(\mathbf{F}, \mathbf{u}_t)$, has full rank. Since $J(\mathbf{F}, \mathbf{u}_t) = \mathbf{A}$, $|\mathbf{A}| = |\mathbf{A} + 0\mathbf{B}| \neq 0$ and the pencil is regular. The analogous argument holds for $\mathbf{B}$ when $\nu_x = 0$. □

As in the simple forcing case, every set of equations that corresponds to a single Jordan block is equivalent to one of the form

$$\mathbf{N}\bar{\mathbf{v}}_a + \mathbf{I}\bar{\mathbf{v}}_b = \bar{\mathbf{g}}(\mathbf{v}) + \bar{\mathbf{f}}(a, b) \tag{4.26}$$

Integrating each equation in turn with respect to b produces an underdetermined set of integral equations, that are implicit in $\mathbf{v}$, of the form

$$\begin{aligned}
\bar{v}_1 &= c_1(a) + \int \left[\bar{g}_1 + \bar{f}_1\right] db \\
\bar{v}_2 &= -bc_1'(a) + c_2(a) + \int \left[\bar{g}_2 + \bar{f}_2 - \int \left[\bar{g}_{1_a} + \bar{f}_{1_a}\right] db\right] db \\
\bar{v}_3 &= \frac{b^2}{2} c_1''(a) - bc_2'(a) + c_3(a) \\
&\qquad + \int \left[\bar{g}_3 + \bar{f}_3 - \int \left[\bar{g}_{2_a} + \bar{f}_{2_a} - \int \left[\bar{g}_{1_{aa}} + \bar{f}_{1_{aa}}\right] db\right] db\right] db \\
&\vdots
\end{aligned} \tag{4.27}$$

Gathering the equations for all Jordan blocks forms a fully determined, implicit set of integral equations in $\mathbf{v}$.

Consider the following system with linear forcing.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

The solution is

$$\begin{aligned}
u(t, x) &= u(0, x) + \int f_1(t, x) dt \\
v(t, x) &= -tu'(0, x) + v(0, x) \\
&\qquad + \int \left(\int f_1(t, x) dt + u(0, x) + f_2(t, x) - \int f_{1_x}(t, x) dt \right) dt
\end{aligned}$$

As expected, existence of a continuous solution to this $\nu_{tot} = 1$ system requires no more than one partial derivative of a forcing function or datum.

For a solution $\mathbf{v}$ to exist pointwise everywhere, each term must exist pointwise everywhere. Because the unrestricted solution is constructed from the functional forms of the forcing functions and data, partial derivatives of a g term represent partial derivatives of the forcing functions and data that make up the dependent variables in that term. The equations and blocks may be coupled through g terms or partial derivative terms in $\mathbf{J}$ and $\mathbf{N}_i$. In either case, the total degeneracy ν_{tot} bounds the number of times an additional order of partial derivative may be implicitly introduced into the solution.

Due to the coupling between different Jordan blocks, the maximum smoothness of $\mathbf{f}$ that may be required for existence of a continuous solution is now equal to ν_{tot}, the sum of the degeneracies of all Jordan blocks. Because the elements of $\mathbf{f}_{i=1,2,3}$ are linear combinations of the original forcing functions $\mathbf{f}$, a sufficient condition for existence of all required derivatives is that each element of $\mathbf{f}$ be ν_{tot}-times differentiable with respect to both t and x. This sufficient smoothness condition is unrelated to the index of the system with respect to any direction in the independent variable space. Again, increasing these sufficient differentiability requirements by one guarantees a smooth solution.

The lower order terms $\mathbf{Cu}$ in the forcing function do not influence the well-posedness of the system. Because the linear forcing term couples the three subsystems of the generalized characteristic form, all three must be considered together.

Theorem 4.3.4 *Assuming that* $(\mathbf{A}, \mathbf{B})$ *forms a regular pencil, the unrestricted solution to* $\mathbf{A}\mathbf{u}_t + \mathbf{B}\mathbf{u}_x + \mathbf{C}\mathbf{u} = \mathbf{0}$ *depends continuously on its data iff the unrestricted solution to* $\mathbf{A}\mathbf{u}_t + \mathbf{B}\mathbf{u}_x = \mathbf{0}$ *depends continuously on its data.*

Proof. Assume the systems are already in canonical form, and consider solution of a single block row of one of the three subsystems for the variables $\mathbf{v}(t, x)$ assigned to it, in terms of the remaining variables $\mathbf{w}(t, x)$. The equations that correspond to this single Jordan block have the form

$$\mathbf{v}_b(a,b) + \mathbf{N}\mathbf{v}_a(a,b) + \mathbf{C}\mathbf{v}(a,b) + \mathbf{C}'\mathbf{w}(a,b) = \mathbf{f}(a,b)$$

At this point, $\mathbf{C}'\mathbf{w}(a,b)$ is simply a vector of unknown functions of a and b, so let $\mathbf{g}(a,b) = \mathbf{f}(a,b) - \mathbf{C}'\mathbf{w}(a,b)$. Taking the Fourier transform of the system produces

$$\hat{\mathbf{v}}_b(b,\omega) + (i\omega\mathbf{N} + \mathbf{C})\,\hat{\mathbf{v}}(b,\omega) = \hat{\mathbf{g}}(b,\omega)$$

The solution is given by

$$\hat{\mathbf{v}}(b,\omega) = e^{-(i\omega\mathbf{N}+\mathbf{C})b}\hat{\mathbf{v}}(0,\omega) + e^{-(i\omega\mathbf{N}+\mathbf{C})b}\int_0^b e^{(i\omega\mathbf{N}+\mathbf{C})s}\hat{\mathbf{g}}(s,\omega)ds$$

By Duhamel's principle, the forced solution may be thought of as a superposition of solutions to the corresponding homogeneous problem, with

$$\hat{\mathbf{v}}^*(0,\omega) = \int_0^b e^{(i\omega\mathbf{N}+\mathbf{C})s}\hat{\mathbf{g}}(s,\omega)ds$$

For the homogeneous problem, it then remains to be shown that

$$||e^{-(i\omega\mathbf{N}+\mathbf{C})b}|| \le C_b \Leftrightarrow ||e^{-(i\omega\mathbf{N})b}|| \le C_b^*$$

for bounded constants C_b and C_b^* independent of ω.

First, assume that $||e^{-(i\omega\mathbf{N})b}|| \le C_b^*$. Then

$$||e^{-(i\omega\mathbf{N}+\mathbf{C})b}|| = ||e^{-i\omega\mathbf{N}b}e^{-\mathbf{C}b}|| \le ||e^{-i\omega\mathbf{N}b}||\;||e^{-\mathbf{C}b}|| \le C_b^*||e^{-\mathbf{C}b}||$$

Now let $C_b^\bullet = ||e^{-\mathbf{C}b}||$ and let $C_b = C_b^*C_b^\bullet$. Then

$$C_b^*||e^{-\mathbf{C}b}|| = C_b$$

Because $\mathbf{C}$ is a constant matrix, $C_b^\bullet$ and thus $C_b^*C_b^\bullet = C_b$ is a function of b only.

For the converse, assume that $||e^{-(i\omega\mathbf{N}+\mathbf{C})b}|| \le C_b$. Then

$$\begin{aligned}||e^{-i\omega\mathbf{N}b}e^{-\mathbf{C}b}|| &\le C_b\\ ||e^{-i\omega\mathbf{N}b}e^{-\mathbf{C}b}||\;||e^{\mathbf{C}b}|| &\le C_b||e^{\mathbf{C}b}||\\ ||e^{-i\omega\mathbf{N}b}|| &\le C_b||e^{\mathbf{C}b}||\end{aligned}$$

and, by an argument similar to the one presented above, the function $C_b^\diamondsuit$ given by $C_b^\diamondsuit = ||e^{Cb}||$ depends only on b, and thus $C_b^* = C_b C_b^\diamondsuit$ is a function of b only.

Because the selection of this first block to be solved is arbitrary, bounds on the unrestricted solution independent of ω hold for every block of $\mathbf{Au}_t + \mathbf{Bu}_x + \mathbf{Cu} = \mathbf{f}(t,x)$ iff they hold for every block of $\mathbf{Au}_t + \mathbf{Bu}_x = \mathbf{f}(t,x)$. $\square$

So well-posed systems with simple forcing always remain well-posed upon addition of linear forcing terms, and ill-posed systems similarly remain ill-posed. The addition of linear forcing terms to a system with simple forcing that is weakly well-posed may make the system strongly ill-posed, however. Consider the following example of such a situation.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_x + \begin{bmatrix} 0 & \epsilon \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.28}$$

The eigenvalues of the coefficient matrix in Fourier space are $\pm(i\epsilon\omega)^{1/2}$. Recall that $\sqrt{ib} = \frac{\sqrt{2b}}{2}(1+i)$, so here the real part of the eigenvalues of the coefficient matrix are $\pm\frac{\sqrt{2\epsilon\omega}}{2}$. The system is therefore strongly ill-posed, while the unforced system ($\epsilon = 0$) is weakly well-posed.

Assuming the unrestricted solution depends continuously on its data, and given a dynamic simulation based on a time evolution method, the same arguments for boundary condition location made in the simple forcing case apply here as well. Data must be specified for hyperbolic blocks with $\tau_i/\rho_i < 0$ on $x = x_2$ and $t = t_0$. Data for the remaining hyperbolic blocks must be specified on $x = x_1$ and $t = t_0$. Data for the differential blocks must be specified on $t = t_0$. Data for an individual parabolic block may be specified on either $x = x_1$ or on $x = x_2$.

4.4 Restricted solutions

Up to this point, only unrestricted solutions have been considered. These unrestricted solutions may depend continuously on their data, or they may be weakly well-posed or strongly ill-posed.

For a restricted solution, some data is used to restrict the space of functions from which the solution is drawn[2], rather than to select a unique member of a space of functions. A perturbation in this data alters the composition of the function space, rather than specifying another unique function that may or may not be near the unique solution to the unperturbed problem. It does not make sense to consider continuous dependence on boundary conditions that contribute to the solution in this manner.

These ideas will be examined in detail here only for the important special case of a 2 × 2 parabolic block.

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{u}_t + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_x + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \mathbf{u} = \mathbf{0} \tag{4.29}$$

Note that $\nu_x = 0$.

Lemma 4.4.1 *A system with linear forcing that consists of a single parabolic block of dimension 2 with $\nu_t = 1$ is ill-posed as an evolution problem in x.*

Proof. Because $\nu_t = 1$, one differentiation with respect to t gives $\mathbf{u}_t$ as a continuous function of $\mathbf{u}$, $\mathbf{u}_x$, t, and x, and by definition the second derivative array equations with respect to t are 1-full. This means that Gauss elimination may be used on the coefficient matrix $\mathcal{A}_2$ of the derivative array equations to produce the form

$$\begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix}$$

where $\mathbf{D}$ is a 2 × 2 diagonal matrix.

For this system,

$$\mathcal{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \frac{\partial}{\partial x} + c_{11} & c_{12} & 0 & 0 \\ c_{21} & \frac{\partial}{\partial x} + c_{22} & 1 & 0 \end{bmatrix}$$

[2] Classically, the term *data* is not even used for any information (initial and/or boundary conditions) that is used in this fashion - see section 2.4.5. In this thesis, however, the term data refers to all initial and boundary conditions, because the analyses developed here make no assumptions about how information is used to construct the solution.

Elimination of the first column produces

$$\mathcal{A}_2^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & c_{12} & 0 & 0 \\ 0 & \frac{\partial}{\partial x} + c_{22} & 1 & 0 \end{bmatrix}$$

At this point it is clear that the matrix is 1-full if and only if $c_{12} \neq 0$; if $c_{12} = 0$, it is impossible to produce both a 2×2 diagonal matrix in the upper lefthand corner and a 2×2 zero matrix in the upper righthand corner. Therefore, since $\mathcal{A}_2$ is 1-full, c_{12} must be nonzero.

Now, take the Fourier transform of the system in t to produce

$$\hat{\mathbf{u}}_x + (i\omega \mathbf{A} + \mathbf{C})\,\hat{\mathbf{u}} = \mathbf{0}$$

The eigenvalues of the coefficient matrix are given by

$$\begin{aligned} \begin{vmatrix} c_{11} - \lambda & c_{12} \\ i\omega + c_{21} & c_{22} - \lambda \end{vmatrix} &= (c_{11} - \lambda)(c_{22} - \lambda) - c_{12}(i\omega + c_{21}) \\ &= \lambda^2 - (c_{11} + c_{22})\lambda + c_{11}c_{22} - c_{12}c_{21} - i\omega c_{12} = 0 \end{aligned}$$

so, by the Quadratic Formula,

$$\lambda = \frac{(c_{11} + c_{22}) \pm \sqrt{(c_{11} + c_{22})^2 - 4(c_{11}c_{22} - c_{12}c_{21} - i\omega c_{12})}}{2}$$

For simplicity, rewrite this expression as

$$\lambda = a \pm \sqrt{b + i\omega c}$$

where $c = 4c_{12}$, and let $d = b + i\omega c$. This complex number d may be given as a magnitude r and phase angle ϕ through the expression $d = r(cos(\phi) + i\sin(\phi))$, where $r = \sqrt{b^2 + \omega^2 c^2}$ and $\phi = \tan^{-1}(\frac{\omega c}{b})$. The square root of d is then given by $\sqrt{r}(\cos(\frac{\phi}{2}) + i\sin(\frac{\phi}{2}))$. Note that

$$\lim_{\omega \to \pm\infty} \tan^{-1}\left(\frac{\omega c}{b}\right) = \pm\frac{\pi}{2}$$

and

$$\lim_{\omega \to \pm\infty} \sqrt{b^2 + \omega^2 c^2} = \omega c$$

Keeping in mind the fact that $c = 4c_{12} \neq 0$, consider what happens in the limit of infinite frequency ω.

$$\begin{aligned}
\lim_{\omega\to\pm\infty} \mathrm{Re}(\lambda) &= \lim_{\omega\to\pm\infty} \mathrm{Re}(a+\sqrt{d}) \\
&= \lim_{\omega\to\pm\infty} a + \sqrt{\sqrt{b^2+\omega^2 c^2}}\cos(\frac{1}{2}\tan^{-1}(\frac{\omega c}{b})) \\
&= \sqrt{\omega c}\cos(\pm\frac{\pi}{4}) \qquad (4.30) \\
&= \pm\frac{\sqrt{2c}}{2}\sqrt{\omega} \\
&\propto \sqrt{\omega}
\end{aligned}$$

Because the real part of the eigenvalues λ have unbounded dependence on ω, the unrestricted solution does not depend continuously on its data. □

Corollary 4.4.2 *A system with linear forcing that consists of a single parabolic block of dimension 2 with $\nu_t = 2$ is weakly well-posed as an evolution problem in x.*

So for a system that consists only of a 2 × 2 parabolic block, if $\nu_t = 2$, there are no dynamic degrees of freedom on $t = 0$, and the problem is weakly well-posed; for $\nu_t = 1$, there is one dynamic degree of freedom on $t = 0$, and the problem is strongly ill-posed. High index in t indicates a weakly well-posed problem in x, while low index in t indicates a strongly ill-posed problem in x.

In either case, $\nu_x = 0$, so specifying two dynamic degrees of freedom on $x = x_1$ determines a unique solution (that may or may not exhibit continuous dependence on that data). However, if $c_{12} \neq 0$, there is a dynamic degree of freedom on $t = 0$.

This apparent contradiction in the number of degrees of freedom on $t = 0$, and the problem of well-posedness, may be resolved by specifying one degree of freedom on $x = x_1$ and the other on $x = x_2$. This restricts the solution to a superposition of sine waves with wavelengths related to the distance $x_2 - x_1$, which can together represent an arbitrary degree of freedom on $t = 0$, as illustrated in the following example.

Example 9 *The heat equation, written as a first order system, is*

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_t + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_x + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \mathbf{0} \qquad (4.31)$$

Consider solutions of the form

$$\mathbf{u}(t,x) = \mathbf{w}(t)\phi(x) \tag{4.32}$$

Substitution into the equations gives

$$\begin{aligned} w_1 &= e^{\lambda^2 t} \\ w_2 &= \lambda e^{\lambda^2 t} \\ \phi &= c_0 e^{-\lambda x} \end{aligned} \tag{4.33}$$

Now consider specification of Dirichlet boundary conditions at $x = 0$ and $x = \pi$, so $\phi(0) = \phi(\pi) = 0$. In this case, we have restricted the solution to a superposition of sines of frequency n/π, where n is any integer, so

$$\begin{aligned} \phi(x) &= \sum_{k=-\infty}^{\infty} c_k sin(kx) \\ &= \frac{-i}{2} \sum_{k=-\infty}^{\infty} c_k e^{ikx} + \frac{i}{2} \sum_{k=-\infty}^{\infty} c_k e^{-ikx} \end{aligned} \tag{4.34}$$

and therefore $\lambda = \pm ik, k \in \mathbb{N}$.

Any reasonable initial condition $w_1(0,x)$ may be represented as an infinite sine series, and will uniquely determine the coefficients c_k.

This restricted solution may be advanced forward in t if it depends continuously on its initial data, which is used to select a unique member of the (restricted) solution space.

Example 10 *Consider the Fourier transform of the heat equation, written as a first order system of the form $\mathbf{A}\hat{\mathbf{u}}_t + \mathbf{B}\hat{\mathbf{u}} = \mathbf{0}$.*

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{u}}_t + \begin{bmatrix} i\omega & 1 \\ 0 & i\omega \end{bmatrix} \hat{\mathbf{u}} = \mathbf{0}$$

The solution to this DAE system may be constructed using the analytical solution presented in chapter 2.3.7.

First, the coefficient matrices $\mathbf{A}$ *and* $\mathbf{B}$ *must be multiplied on the left by a matrix of the form* $(\mathbf{B}-\lambda\mathbf{A})^{-1}$*. Because* $\mathbf{B}$ *is invertible, the simplest choice is to let* $\lambda = 0$ *so that* $(\mathbf{B}-\lambda\mathbf{A})^{-1} = \mathbf{B}^{-1}$*.*

$$\hat{\mathbf{A}} = \mathbf{B}^{-1}\mathbf{A} = \begin{bmatrix} -\frac{i}{\omega} & \frac{1}{\omega^2} \\ 0 & -\frac{i}{\omega} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\omega^2} & 0 \\ -\frac{i}{\omega} & 0 \end{bmatrix}$$

$$\hat{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Now, note that the eigenvectors of $\hat{\mathbf{A}}$ *that correspond to nonzero eigenvalues, followed by eigenvectors that correspond to zero eigenvalues, form a matrix* $\mathbf{T}$ *that takes both* $\hat{\mathbf{A}}$ *and* $\hat{\mathbf{B}}$ *to their desired diagonal forms. The eigenvalues of* $\hat{\mathbf{A}}$ *are given by*

$$\begin{vmatrix} \frac{1}{\omega^2} - \lambda & 0 \\ -\frac{i}{\omega} & -\lambda \end{vmatrix} = \lambda^2 - \frac{1}{\omega^2}\lambda = 0 \;\;\Rightarrow\;\; \lambda = \frac{1}{\omega^2},\, 0$$

so the eigenvectors are given by

$$\hat{\mathbf{A}}\mathbf{x}_1 = \frac{1}{\omega^2}\mathbf{x}_1 \;\;\Rightarrow\;\; \mathbf{x}_1 = \begin{bmatrix} 1 \\ -i\omega \end{bmatrix}$$

$$\hat{\mathbf{A}}\mathbf{x}_2 = \mathbf{0} \;\;\Rightarrow\;\; \mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and finally $\mathbf{T}$ *and* $\mathbf{T}^{-1}$ *are*

$$\mathbf{T} = \begin{bmatrix} 1 & 0 \\ -i\omega & 1 \end{bmatrix} \qquad \mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 \\ i\omega & 1 \end{bmatrix}$$

Now, $\hat{\mathbf{A}}^D$ *can be calculated from* $\mathbf{T}^{-1}\hat{\mathbf{A}}\mathbf{T}$*. First,*

$$\mathbf{T}^{-1}\hat{\mathbf{A}}\mathbf{T} = \begin{bmatrix} 1 & 0 \\ i\omega & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\omega^2} & 0 \\ -\frac{i}{\omega} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -i\omega & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\omega^2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -i\omega & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\omega^2} & 0 \\ 0 & 0 \end{bmatrix}$$

so by the definition of $\hat{\mathbf{A}}^D$*,*

$$\mathbf{T}^{-1}\hat{\mathbf{A}}^D\mathbf{T} = \begin{bmatrix} \omega^2 & 0 \\ 0 & 0 \end{bmatrix}$$

Multiplication on the left by $\mathbf{T}$ *and on the right by* $\mathbf{T}^{-1}$ *then gives* $\hat{\mathbf{A}}^D$ *directly.*

$$\begin{aligned}\hat{\mathbf{A}}^D &= \mathbf{T}\left(\mathbf{T}^{-1}\hat{\mathbf{A}}\mathbf{T}\right)\mathbf{T}^{-1}\\ &= \begin{bmatrix} 1 & 0 \\ -i\omega & 1 \end{bmatrix}\begin{bmatrix} \omega^2 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ i\omega & 1 \end{bmatrix} = \begin{bmatrix} \omega^2 & 0 \\ -i\omega^3 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ i\omega & 1 \end{bmatrix} = \begin{bmatrix} \omega^2 & 0 \\ -i\omega^3 & 0 \end{bmatrix}\end{aligned}$$

The products $\hat{\mathbf{A}}^D\hat{\mathbf{B}}$ *and* $\hat{\mathbf{A}}\hat{\mathbf{A}}^D$ *appear in the analytical solution, and are given by*

$$\hat{\mathbf{A}}^D\hat{\mathbf{B}} = \begin{bmatrix} \omega^2 & 0 \\ -i\omega^3 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \omega^2 & 0 \\ -i\omega^3 & 0 \end{bmatrix}$$

$$\hat{\mathbf{A}}\hat{\mathbf{A}}^D = \begin{bmatrix} \frac{1}{\omega^2} & 0 \\ -\frac{i}{\omega} & 0 \end{bmatrix}\begin{bmatrix} \omega^2 & 0 \\ -i\omega^3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -i\omega & 0 \end{bmatrix}$$

The solution is then

$$\hat{\mathbf{u}}(t,\omega) = e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D\hat{\mathbf{u}}(0,\omega)$$

Now, the question is whether or not the norm of the solution at some later t *can be bounded by a function* C_t *independent of* ω *and the norm of the initial data. In other words, is there a* C_t *such that*

$$||\hat{\mathbf{u}}(t,\omega)|| \leq C_t||\hat{\mathbf{u}}(0,\omega)||$$

for all $\omega \in \mathbb{R}$ *and* $t > 0$*?*

Taking the norm of both sides of the solution, it is clear that

$$||\hat{\mathbf{u}}(t,\omega)|| = ||e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D\hat{\mathbf{u}}(0,\omega)|| \leq ||e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D||\;||\hat{\mathbf{u}}(0,\omega)||$$

and the solution will depend continuously on its data if there exists some bounded function C_t *that depends only on* t *for which*

$$||e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D|| \leq C_t$$

for all $\omega \in \mathbb{R}$*.*

First, by the series definition of the exponential of a matrix,

$$\begin{aligned} e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t} &= \mathbf{I} + (-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t) + \frac{1}{2!}(-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t)^2 + \frac{1}{3!}(-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t)^3 + \dots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -\omega^2 t & 0 \\ i\omega^3 t & 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\omega^4 t^2 & 0 \\ -\frac{1}{2}i\omega^5 t^2 & 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{6}\omega^6 t^3 & 0 \\ \frac{1}{6}i\omega^7 t^3 & 0 \end{bmatrix} + \dots \\ &= \begin{bmatrix} 1 + \sum_{j=1}^{\infty} \frac{1}{j!}(-\omega^2 t)^j & 0 \\ -i\omega \sum_{j=1}^{\infty} \frac{1}{j!}(-\omega^2 t)^j & 1 \end{bmatrix} \end{aligned}$$

Recall that

$$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \dots = 1 + \sum_{j=1}^{\infty} \frac{1}{j!}x^j$$

so clearly the upper lefthand element of $e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}$ *is simply an exponential:*

$$1 + \sum_{j=1}^{\infty} \frac{1}{j!}(-\omega^2 t)^j = e^{-\omega^2 t}$$

Adding zero, in the form $i\omega - i\omega$, *to the lower left entry of the matrix allows that entry to also be written as an exponential.*

$$i\omega - i\omega - i\omega \sum_{j=1}^{\infty} \frac{1}{j!}(-\omega^2 t)^j = -i\omega(-1 + e^{-\omega^2 t}) = i\omega(1 - e^{-\omega^2 t})$$

Therefore,

$$e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t} = \begin{bmatrix} e^{-\omega^2 t} & 0 \\ i\omega(1 - e^{-\omega^2 t}) & 1 \end{bmatrix}$$

and the product $e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D$ *may be calculated as follows.*

$$e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D = \begin{bmatrix} e^{-\omega^2 t} & 0 \\ i\omega(1 - e^{-\omega^2 t}) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -i\omega & 0 \end{bmatrix} = \begin{bmatrix} e^{-\omega^2 t} & 0 \\ -i\omega e^{-\omega^2 t} & 0 \end{bmatrix}$$

Can the norm of this matrix be shown to be less than some function of t *only for all values of* ω*?*

Recall that the norm of a matrix $\mathbf{M}$ *is equal to the square root of the eigenvalue of largest magnitude of* $\mathbf{M}^T\mathbf{M}$. *Let*

$$\mathbf{M} = \begin{bmatrix} e^{-\omega^2 t} & 0 \\ -i\omega e^{-\omega^2 t} & 0 \end{bmatrix}$$

so

$$\mathbf{M}^T\mathbf{M} = \begin{bmatrix} e^{-\omega^2 t} & -i\omega e^{-\omega^2 t} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e^{-\omega^2 t} & 0 \\ -i\omega e^{-\omega^2 t} & 0 \end{bmatrix} = \begin{bmatrix} (1-\omega^2)e^{-2\omega^2 t} & 0 \\ 0 & 0 \end{bmatrix}$$

and the eigenvalue of largest magnitude is clearly $\lambda_{max} = (1-\omega^2)e^{-2\omega^2 t}$*, so* $\sqrt{\lambda_{max}} = (1-\omega^2)^{1/2}e^{-\omega^2 t}$.

For $-1 \leq \omega \leq 1$*,* λ_{max} *is positive or zero, so* $\sqrt{\lambda_{max}}$ *is a real number. Therefore*

$$|\sqrt{\lambda_{max}}| = (1-\omega^2)^{1/2}e^{-\omega^2 t} \quad \textit{for} \quad -1 \leq \omega \leq 1$$

Because $(1-\omega^2)^{1/2} \leq 1$ *and* $e^{-\omega^2 t} \leq 1$ *for* $-1 \leq \omega \leq 1$ *and* $t > 0$*, the maximum value of the product of these two terms is also one, and thus*

$$||e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D|| = |\sqrt{\lambda_{max}}| \leq 1 \quad \textit{for} \quad -1 \leq \omega \leq 1$$

For $\omega < -1$ *or* $\omega > 1$*,* λ_{max} *is negative and real, so* $\sqrt{\lambda_{max}}$ *is a pure imaginary number. For* $b > 0$*, because* $\sqrt{-b} = i\sqrt{b}$ *and* $|i\sqrt{b}| = \sqrt{b}$*, clearly* $|\sqrt{-b}| = \sqrt{b}$ *so*

$$|\sqrt{\lambda_{max}}| = (\omega^2-1)^{1/2}e^{-\omega^2 t} \quad \textit{for} \quad \omega < -1, \ \omega > 1$$

In the limit $\omega \rightarrow \pm\infty$*, this quantity goes to zero for all* $t > 0$*. Also, as* ω *approaches* ± 1*, the quantity goes to zero. It is a continuous function; its derivative with respect to* ω *is given by*

$$\frac{\partial}{\partial\omega}|\sqrt{\lambda_{max}}| = \left((\omega^2-1)^{-1/2} - 2t(\omega^2-1)^{1/2}\right)\omega e^{-\omega^2 t}$$

and exists for all $\omega < -1$ *and* $\omega > 1$ *with* $t > 0$*. The maximum value of* $|\sqrt{\lambda_{max}}|$ *will occur at the value of* ω *where the derivative vanishes, given by*

$$\begin{aligned} 0 &= \left((\omega^2-1)^{-1/2} - 2t(\omega^2-1)^{1/2}\right)\omega e^{-\omega^2 t} \\ (\omega^2-1)^{-1/2} &= 2t(\omega^2-1)^{1/2} \\ \omega^2 &= 1 + \frac{1}{2t} \end{aligned}$$

Substituting this value into the expression for $|\sqrt{\lambda_{max}}|$ *gives*

$$||e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D|| = |\sqrt{\lambda_{max}}| \leq \left(\left(1 + \frac{1}{2t}\right) - 1\right)^{1/2} e^{-(1+\frac{1}{2t})t} = \frac{1}{\sqrt{2t}}e^{-(t+1/2)}$$

for all $\omega < -1$ and $\omega > 1$ with $t > 0$.

Because the bounds given by

$$\begin{aligned} ||e^{-\hat{\mathbf{A}}^D\hat{\mathbf{B}}t}\hat{\mathbf{A}}\hat{\mathbf{A}}^D|| &\leq 1 \quad for \quad -1 \leq \omega \leq 1 \\ &\leq \frac{1}{\sqrt{2t}}e^{-t+1/2} \quad for \quad \omega < -1, \ \omega > 1 \end{aligned}$$

are finite and independent of ω for all $t > 0$ and all $\omega \in \mathbb{R}$, the restricted solution depends continuously on its data.

So, while the unrestricted solution to a parabolic block of dimension 2 may be strongly ill-posed because it does not depend continuously on its initial data in x, there may still be a restricted solution that is well-posed as an evolution problem in t.

The same arguments may be applied to a degenerate differential block. By lemma 4.4.1, a degenerate differential block of dimension 2 with $\nu_x = 2$ depends continuously on its initial data. By the same lemma, if $\nu_x = 1$, it is ill-posed as an evolution problem in t. However, because the overall solution method for a dynamic simulation is assumed to be evolutionary in t, it is in general not permissible to enforce data simultaneously at two different values of t.

4.5 Systems with a singular coefficient matrix pencil

A system for which the coefficient matrix pair $(\mathbf{A}, \mathbf{B})$ does not form a regular pencil, but that is equivalent to an algebraic system coupled to a PDE with a regular coefficient matrix pencil

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}_t + \begin{bmatrix} \mathbf{B}_{11} & \mathbf{0} \\ \mathbf{B}_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}_x = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix} \qquad (4.35)$$

where $dim(\mathbf{C}_{22}) = n - r$, $r = max_{\lambda \in \mathbb{R}}(rank(\mathbf{A} + \lambda\mathbf{B}))$, with $(\mathbf{A}_{11}, \mathbf{B}_{11})$ regular and $\mathbf{C}_{22}$ invertible, may be handled in the same manner as one with a regular pencil. Because the first block row involves only $\mathbf{u}_1$, it may be considered independently of the second block row. Again assuming a dynamic simulation based on a time evolution method, the first block row provides the same information regarding dependence on and location of data given by characteristic analysis in the regular coefficient matrix pencil case. Once the first block row is solved for $\mathbf{u}_1$, no additional data is required to uniquely determine $\mathbf{u}_2$. Therefore let the second block row be called the **algebraic part** of the system.

Lemma 4.5.1 $\mathbf{u}_2$ *depends on at most* $\nu_{tot} + 1$ *partial derivatives of* $\mathbf{f}_1$*, where* ν_{tot} *is the total of the degeneracies of all of the blocks in the canonical form of the first block row.*

Proof. The algebraic variables $\mathbf{u}_2$ are given by

$$\mathbf{u}_2 = \mathbf{C}_{22}^{-1} \left[\mathbf{A}_{21}\mathbf{u}_{1_t} + \mathbf{B}_{21}\mathbf{u}_{1_x} - \mathbf{C}_{21}\mathbf{u}_1 - \mathbf{f}_2 \right] \tag{4.36}$$

Because $\mathbf{u}_1$ is the solution to a system with linear forcing, it depends on at most ν_{tot} partial derivatives of $\mathbf{f}_1$. By inspection $\mathbf{u}_2$ depends on at most $\nu_{tot} + 1$ partial derivatives of $\mathbf{f}_1$. □

A differential system that is equivalent to an algebraic system may also be coupled to a regular PDE and handled in the same way. Let $\mathbf{N}_1$ and $\mathbf{N}_2$ be two conforming nonzero nilpotent matrices, both either strictly upper triangular or strictly lower triangular.

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{N}_1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}_t + \begin{bmatrix} \mathbf{B}_{11} & \mathbf{0} \\ \mathbf{B}_{21} & \mathbf{N}_2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}_x = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix} \tag{4.37}$$

Because the algebraic subsystem includes coupling via partial derivatives of the algebraic variables $\mathbf{u}_2$, there may be additional dependence on derivatives of the forcing functions. Let ν_1 and ν_2 be the nilpotencies of $\mathbf{N}_1$ and $\mathbf{N}_2$, respectively.

Lemma 4.5.2 $\mathbf{u}_2$ *depends on at most* $\nu_{tot} + \nu_1 + 1$ *partial derivatives of* $\mathbf{f}_1$ *and* ν_1 *partial derivatives of* $\mathbf{f}_2$ *with respect to* t*, and on at most* $\nu_{tot}+\nu_2+1$ *partial derivatives of* $\mathbf{f}_1$ *and* ν_2 *partial derivatives of* $\mathbf{f}_2$ *with respect to* x*.*

Proof. Let $k = \max(\nu_1, \nu_2)$ and $\mathbf{N} = \mathbf{N}_1\frac{\partial}{\partial t} + \mathbf{N}_2\frac{\partial}{\partial x}$. The algebraic variables $\mathbf{u}_2$ are given by

$$\mathbf{u}_2 = \left(\sum_{i=0}^{k} \mathbf{N}^i\right)(\mathbf{A}_{21}\mathbf{u}_{1_t} + \mathbf{B}_{21}\mathbf{u}_{1_x} - \mathbf{C}_{21}\mathbf{u}_1 - \mathbf{f}_2) \tag{4.38}$$

Because $\mathbf{u}_1$ is the solution to a system with linear forcing, it depends on at most ν_{tot} partial derivatives of $\mathbf{f}_1$. The lemma follows by inspection. □

The importance of eliminating the algebraic variables $\mathbf{u}_2$ from the first block row prior to characteristic analysis is illustrated by the following simple example.

Example 11 *Consider the question of proper data for the following simple system.*

$$\begin{aligned} u_x + 3v &= 0 \\ u_t + v &= 0 \end{aligned} \tag{4.39}$$

Clearly only one equation will give rise to a constant of integration. Suppose the second equation is assigned to v. The first equation then apparently determines u up to a constant of integration that may depend on t. However, the occurrence of the algebraic variable v in the first equation introduces another partial derivative of u, so upon elimination of v it becomes clear that u is a one-way wave travelling with speed $-1/3$:

$$u_t - \frac{1}{3}u_x = 0 \tag{4.40}$$

An important special case is systems that contain one or more strictly algebraic equations. An algebraic equation constrains the dependent variables on *every* surface in the independent variable space, so a system that contains an algebraic equation may be viewed as one for which every surface is characteristic. This corresponds to $\mathbf{A}_{21} = \mathbf{B}_{21} = \mathbf{0}$ in the form considered above (4.35). In such a case, elimination of the algebraic variables from the first block row will not produce additional derivative terms, and so is not necessary.

If an algebraic equation is differentiated once with respect to time, it becomes an ordinary differential equation. This is an interior partial differential equation on surfaces of the form $x = constant$, such as the domain boundaries. In other words, differentiation with respect to t transforms an algebraic equation, which constrains the solution on all surfaces, to one that constrains the solution on domain boundaries of the form $x = constant$. If one is interested in the equations that partially determine the solution $\mathbf{u}$ on a domain boundary, the original and differentiated algebraic equations are equivalent.

Definition 4.5.3 *A variable u is* **x-algebraic** *iff no partial derivative of u with respect to x appears in the system.*

Definition 4.5.4 *A variable u is* **x-differential** *iff it is not x-algebraic.*

Definition 4.5.5 *An equation $f(\mathbf{u}) = 0$ is* **x-algebraic** *iff no partial derivatives with respect to x appear in it.*

Definition 4.5.6 *An equation $f(\mathbf{u}) = 0$ is* **x-differential** *iff it is not x-algebraic.*

Lemma 4.5.7 $\left(\begin{bmatrix}\mathbf{A}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{B}\\ \mathbf{C}\end{bmatrix}\right)$ *regular* $\Rightarrow$ $\left(\begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}, \begin{bmatrix}\mathbf{B}\\ \mathbf{0}\end{bmatrix}\right)$ *regular.*

Proof. $\left(\begin{bmatrix}\mathbf{A}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{B}\\ \mathbf{C}\end{bmatrix}\right)$ regular $\Rightarrow$ $\exists\, \lambda \in \mathbb{R}$ such that $\begin{vmatrix}\mathbf{A} + \lambda\mathbf{B}\\ \lambda\mathbf{C}\end{vmatrix} \neq 0$. Clearly $\lambda \neq 0$. Let $\mathbf{C} \in \mathbb{R}^{m \times n}$. Multiply the last m rows of $\begin{vmatrix}\mathbf{A} + \lambda\mathbf{B}\\ \lambda\mathbf{C}\end{vmatrix}$ by $\frac{1}{\lambda}$. Then $\begin{vmatrix}\mathbf{A} + \lambda\mathbf{B}\\ \mathbf{C}\end{vmatrix} = \left(\frac{1}{\lambda}\right)^m \begin{vmatrix}\mathbf{A} + \lambda\mathbf{B}\\ \lambda\mathbf{C}\end{vmatrix} \neq 0 \;\Rightarrow\; \left(\begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}, \begin{bmatrix}\mathbf{B}\\ \mathbf{0}\end{bmatrix}\right)$ is regular. $\square$

Let $V^{(Dt)}$ and $V^{(At)}$ be the set of t-differential and t-algebraic variables respectively, and let $V^{(Dx)}$ and $V^{(Ax)}$ be the sets of x-differential and x-algebraic variables. Let $E^{(Dt)}$ and $E^{(At)}$ be the sets of t-differential and t-algebraic equations, and let $E^{(Dx)}$ and $E^{(Ax)}$ be the sets of x-differential and x-algebraic equations.

Theorem 4.5.8 *If $\nu_t = 1$, $J(E^{(Dt)}, V^{(Dt)})$ and $J(E^{(At)}, V^{(At)})$ have full row rank, and $E^{(At)} \cap E^{(Dx)} = \phi$, then differentiating the algebraic equations once with respect to t produces a first order system with a regular coefficient matrix pencil.*

Proof. Because $E^{(At)} \cap E^{(Dx)} = \phi$, differentiating every member of $E^{(At)}$ once with respect to time produces a first order system. Since $\nu_t = 1$ and both $J(E^{(Dt)}, V^{(Dt)})$ and $J(E^{(At)}, V^{(At)})$ have full rank, differentiating the t-algebraic equations once produces a system for which $\nu_t = 0$, and by lemma 4.3.3, the coefficient matrix pencil must be regular. □

Corollary 4.5.9 *If $\nu_x = 1$, both $J(E^{(Dx)}, V^{(Dx)})$ and $J(E^{(Ax)}, V^{(Ax)})$ have full row rank, and $E^{(Ax)} \cap E^{(Dt)} = \phi$, then differentiating the algebraic equations once with respect to t produces a first order system with a regular coefficient matrix pencil.*

Proof. One differentiation with respect to x produces a regular pencil, by an argument identical to that for differentiation by t in the proof of theorem 4.5.8. Lemma 4.5.7 guarantees that differentiation by t rather than x also produces a regular pencil. □

4.6 Quasilinear systems

While the analyses and automation methods presented in this chapter deal strictly with linear systems, an important issue is their applicability to semilinear and quasilinear systems. The index of a quasilinear system is a local property in $(t, x, \mathbf{u})$-space [54]. Index analysis based on structural algorithms may be applied to quasilinear and nonlinear systems, subject to the considerations involving numerical singularities mentioned above, which in the quasilinear case may occur only locally. Structural algorithms have enjoyed some success when applied to nonlinear DAEs in chemical engineering literature [27], although examples where they fail are also well known [71].

Characteristic analysis of semilinear and quasilinear systems may be automated by freezing the coefficients at some nominal value $(\mathbf{u}_0, x_0, t_0)$ of interest. For quasilinear hyperbolic systems, the boundary condition requirements for the frozen system will be valid only locally [18], and one expects the same to hold for systems that contain hyperbolic and differential subsystems. For problems that contain a parabolic subsystem, additional assumptions on the variables associated with the infinite speed characteristics will undoubtedly be required, because the boundary condition problem for these systems is inherently nonlocal.

An important question is whether continuous dependence of a frozen coefficient linearization on its data implies the same property in the original system. A formal linearization produces the system that governs small perturbations about a nominal value; this typically introduces lower-order terms [44]. Quasilinear hyperbolic systems for which the frozen coefficient system depends continuously on its data may be perturbed by lower order terms and retain continuous dependence; formal linearizations that describe small perturbations about the same nominal value at which the system was frozen thus also depend continuously on their data, and so the original system is said to be continuously dependent on its data at that point [44]. By analogy using theorem 4.3.4, continuous dependence of a frozen coefficient system on its data should imply the same property in all formal linearizations of the original system around $(\mathbf{u}_0, x_0, t_0)$, and given proper initial and boundary conditions, the original quasilinear system might then be considered well-posed at $(\mathbf{u}_0, x_0, t_0)$.

4.7 The degeneracy and perturbation index

Campbell and Marszalek deal primarily with restricted solutions, where all boundary conditions are used to determine the function space, and initial data is used to select a unique solution. The perturbation index is by definition the highest order derivative of either initial data or forcing functions that appears in the restricted solution.

The approach presented here deals primarily with unrestricted solutions, built on a characteristic interpretation of fairly general linear, first order systems. The degeneracy of the system is shown to provide an upper bound on the order of derivatives of data and forcing functions that can appear in the unrestricted solutions. Restricted solutions are considered only for degenerate parabolic parts of first order systems.

The consistent Cauchy data problem is a (typically underdetermined) interior partial differential system in the dependent variables and their exterior partial derivatives; e.g., for a system of first order in t, the consistent Cauchy data problem on $(t = t_0)$ is some underdetermined PDE in the $2n$ quantities $\mathbf{u}_t(0, x)$ and $\mathbf{u}(0, x)$. A unique solution may require specification of some of those quantities over the entire initial surface $t = 0$; these arbitrary specifications are called dynamic degrees of freedom on $t = 0$. Because in general it is an interior partial differential system, depending on what dynamic degrees of freedom are specified, additional data on lower dimensional subsurfaces within $t = 0$ may be required in order to determine a unique solution to the consistent initialization problem. Also, the initial and boundary data must agree at all points of intersection, or a corner singularity will produce a discontinuity.

Example 12 *Consider the following problem [12].*

$$\begin{aligned} u_t + v_{xx} &= f_1(t, x) \\ v_{xx} - w &= f_2(t, x) \\ w_t &= f_3(t, x) \end{aligned} \tag{4.41}$$

Let the domain be $0 \le x \le L$, $t \ge 0$, *and let the initial and boundary conditions be*

$$\begin{aligned} v(t, 0) = v(t, L) &= 0 \\ u(0, x) &= u_0(x) \end{aligned} \tag{4.42}$$

along with either

$$v(0, x) = v_0(x) \tag{4.43}$$

or

$$w(0, x) = w_0(x) \tag{4.44}$$

The differentiation index with respect to t of this system is 1. There are no implicit constraints on Cauchy data on $t = 0$. *Let* $y = u_t$ *and* $z = w_t$. *The equations that constrain consistent Cauchy data are*

$$\begin{aligned} y + v_{xx} &= f_1(t, x) \\ v_{xx} - w &= f_2(t, x) \\ z &= f_3(t, x) \end{aligned} \tag{4.45}$$

so there are two dynamic degrees of freedom.

Consider the two specifications presented in [12]. Case I will be

$$\begin{aligned} u(0,x) &= u_0(x) \\ w(0,x) &= w_0(x) \end{aligned} \tag{4.46}$$

and Case II will be

$$\begin{aligned} u(0,x) &= u_0(x) \\ v(0,x) &= v_0(x) \end{aligned} \tag{4.47}$$

Both consistent Cauchy problems are interior PDEs on $t = 0$. Specifically they form a DAE in x. The differentiation index of Case I with respect to x is 1. It has two dynamic degrees of freedom that may be specified on a point within $t = 0$. All variables except for v and v_x are uniquely determined. The boundary data for the original problem provides the two dynamic degrees of freedom needed for this second lower dimensional consistent Cauchy data problem.

The differentiation index of Case II with respect to x is 3. No dynamic degrees of freedom exist on surfaces of the form $t = 0, x = k$, so there is no lower dimensional consistent Cauchy data problem for Case II. The consistent Cauchy data problem for Case I or Case II would have been a strictly algebraic system anyway.

In Case II, the second derivative of the data $v_0(x)$ is clearly used to determine $w(0,x)$, so the perturbation index of the original problem is 3. In Case I, no derivatives of data appear in the solution, so the perturbation index is 1.

This example highlights a fundamental difference between the differentiation and perturbation index analyses. The former approach performs differentiation index analysis with respect to the normal direction to the initial hyperplane, in order to determine the number of dynamic degrees of freedom over the entire initial hyperplane. Specification of those dynamic degrees of freedom yields a new problem in one fewer independent variables. Different specifications lead to different new problems.

The new problem is treated as distinct. Index analysis may again be used to determine the number of dynamic degrees of freedom on a particular hyperplane within this new, lower dimensionality independent variable space. Repeated application of differentiation index analysis on consistent Cauchy data problems and subproblems can thus capture the different smoothness requirements that appear at different stages of the solution.

The latter approach considers all such lower dimensional problems together with the original. The perturbation index reflects the derivatives of data that appear in the solution of the original consistent Cauchy data problem together with any derivatives that appear in consistent Cauchy data subproblems.

Contrast these approaches with the calculation of the degeneracy for this example. The degeneracy of a linear system gives an upper bound on the order of derivatives and forcing functions that appear in the solution. The system may be reduced to first order by introducing a new variable $s = v_x$ to produce a linear system with linear forcing.

$$\begin{aligned}\begin{bmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}u\\v\\w\\s\end{bmatrix}_t+\begin{bmatrix}0&0&0&1\\0&1&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}\begin{bmatrix}u\\v\\w\\s\end{bmatrix}_x&\\+\begin{bmatrix}0&0&0&0\\0&0&0&-1\\0&0&-1&0\\0&0&0&0\end{bmatrix}\begin{bmatrix}u\\v\\w\\s\end{bmatrix}&=\begin{bmatrix}f_1(t,x)\\0\\f_2(t,x)\\f_3(t,x)\end{bmatrix}\end{aligned}\tag{4.48}$$

The canonical form of the system is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ v \\ s \end{bmatrix}_t + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ w \\ v \\ s \end{bmatrix}_x + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ v \\ s \end{bmatrix} = \begin{bmatrix} f_1(t,x) - f_2(t,x) \\ f_3(t,x) \\ 0 \\ f_2(t,x) \end{bmatrix} \tag{4.49}$$

The degeneracy of the system is therefore zero, which implies that no derivatives of data or forcing functions occur in the solution.

This result may be reconciled with the differentiation and perturbation index analyses by recalling that the degeneracy assumes an unrestricted, characteristic-like solution. Indeed, if data u and w are given on $t = 0$, and for v and s on $x = 0$, the solution does not depend on any derivatives of that data or the forcing functions.

$$\begin{aligned} w(t,x) &= \int_0^t f_3(\tau,x)d\tau + w(0,x) \\ u(t,x) &= \int_0^t \left[f_1(\tau,x) - f_2(\tau,x) - \int_0^\tau f_3(\eta,x)d\eta + w(0,x) \right] d\tau + u(0,x) \\ s(t,x) &= \int_0^x \left[f_2(t,\chi) + \int_0^t f_3(\tau,\chi)d\tau + w(0,\chi) \right] d\chi + s(t,0) \\ v(t,x) &= \int_0^x \left[\int_0^\chi \left[f_2(t,\psi) + \int_0^t f_3(\tau,\psi)d\tau + w(0,\psi) \right] d\psi \right] d\chi + v(t,0) \end{aligned} \tag{4.50}$$

Specifically, the degeneracy analysis *inherently assumes* that the initial conditions will be given for u and w. With this specification, the index analyses reflect the bound given by the degeneracy. The combined application of degeneracy and the recursive Cauchy analysis does, however, yield the same information as the perturbation index.

Note the subtle problem with the specification used in Case II. Specifically, in this consistent Cauchy data subproblem, no dynamic degrees of freedom exist on points within $t = 0$. The boundary conditions for the original problem are therefore either inconsistent or redundant with this second consistent Cauchy data problem.

Chapter 5

Implementation and Examples

5.1 Implementation

The goal of this work is to automate the analyses of the previous chapters as much as possible. In particular, determination of the index, degeneracy, characteristic directions, and variables associated with the subsystems of the canonical form will allow a simulator to verify initial and boundary conditions, identify systems of high index with respect to the evolution variable t, and detect some ill-posed systems.

Difficulties with direct calculation of the canonical form of a DAE [9] and a desire to develop methods that may be used for nonlinear problems have led to the development of structural index algorithms [45, 63]. These algorithms work with the occurrence information to determine the minimum number of differentiations required to produce a low index (zero or one) system. It is well known that DAEs of high index due to numerical singularities may escape detection by structural algorithms. Recent work [71] has highlighted the fact that structural algorithms may also *overestimate* the number of differentiations required to produce a low index system. However, the low computational cost of these algorithms and their applicability to nonlinear and large, sparse systems allows them to be used with considerable success in practical applications[1].

[1] If new algorithms emerge that provably perform this analysis properly, then they can be applied directly and the answer will be unambiguous.

A second algorithm, called the method of dummy derivatives [56] has been used successfully in conjunction with Pantelides' algorithm to automatically generate a low index system that is mathematically equivalent to the original system and explicitly preserves all constraints. From this dummy reformulation of the original system, one may obtain the dynamic degrees of freedom, which is equal to the number of differential variables. Note that this number may be correct even in the case where the number of differentiations has been overstated by the structural algorithm.

Both algorithms may be applied in an extremely straightforward manner to PDEs. The index with respect to t, for example, is determined by considering all interior partial differential operators together with algebraic operators. Whether the calculations would be done using Laplace transforms or operator-valued coefficient matrices, the incidence matrix for t-algebraic occurrences of the dependent variables is formed by simply merging the incidence matrices for $\mathbf{u}_x$ and $\mathbf{u}$. Once this has been done, the two algorithms will (in the absence of numerical singularities) produce an equivalent system of index 0 or 1 with respect to t that reflects the true number of t-differential variables. The number of initial conditions required in order to determine a unique solution is equal to the number of t-differential variables in the t-dummy reformulation.

The most basic necessary condition for well-posedness of a linear system is the regularity condition of Campbell and Marszalek [12]. In order for it to satisfy the regularity condition, the system must be an output set. In order for the system to have an output set, it must have a transversal with respect to all occurrences of the dependent variables and their partial derivatives. This chain of implications reveals that existence of a transversal is a necessary condition for well-posedness. Pantelides' algorithm checks for this transversal as a preprocessing step that guarantees the algorithm has finite termination. A numerical, rather than structural, check of regularity for systems with simple forcing will be presented below.

Routines that calculate the generalized eigenvalues and their degeneracies for regular coefficient matrix pairs are readily available [33, 21]. If any generalized eigenvalues are complex, the system is ill-posed. Otherwise, if the degeneracy of the system is zero, theorems 4.2.6 and 4.3.4 guarantee that the solution depends continuously on its data. If the degeneracy of the system is nonzero but the forcing is simple, the system is weakly well-posed. For linear forcing and nonzero degeneracy, it is not in general possible at present to distinguish between weakly well-posed and strongly ill-posed systems.

Index analysis may be used to identify the total number of boundary conditions required to determine a unique solution. Just as index analysis with respect to t gives the number of dynamic degrees of freedom on surfaces of the form $t = const$, index analysis with respect to x gives the number of dynamic degrees of freedom on surfaces of the form $x = const$. In a dynamic simulation with t as the evolution variable, all such degrees of freedom on surfaces of the form $t = const$ must be specified as initial conditions, while dynamic degrees of freedom on surfaces of the form $x = const$ may be specified on either $x = x_1$ or $x = x_2$.

The distribution of these boundary conditions between the boundaries $x = x_1$ and $x = x_2$ may be ascertained from the generalized eigenvalues. Each block in the hyperbolic subsystem was shown to be equivalent to an ODE along a particular direction in the (x, t) plane, given by $dx/dt = \tau_i/\rho_i$. Because a dynamic simulation in t is assumed, data provided at t_2 may not be used to specify a unique solution at $t_1 < t_2$, so initial conditions for these ODEs must be provided as boundary conditions on $x = a$ for ODEs along $dx/dt > 0$, and as boundary conditions on $x = b$ for ODEs along $dx/dt < 0$.

Blocks in the parabolic subsystem are equivalent to ODEs in x, or along the direction $dt/dx = 0$. An initial condition for such an ODE may in general be given at either domain boundary in x. In particular, a parabolic block of dimension 1 requires a boundary condition at either $x = a$ or $x = b$.

If the only blocks with nonzero degeneracy are part of the parabolic subsystem and of dimension 2, and the index of the system with respect to t is 1, lemma 4.4.1 guarantees that the solution to the parabolic blocks will not depend continuously on their data if that data is enforced at a single end of the domain boundary in x. Example 10 shows that such a problem may still be well-posed as an evolution problem in t if one boundary condition is enforced at each end of the domain boundary in x for every parabolic block of degeneracy 1.

By the same approach but with the roles of t and x reversed, if the only blocks with nonzero degeneracy are part of the differential subsystem and the index of the system with respect to x is 1, lemma 4.4.1 guarantees that the solution will not depend continuously on its data if that data is enforced at a single surface. As an evolution problem in t, the problem is therefore ill-posed.

It is possible to move beyond simply counting the number of required boundary conditions and to identify the information that those boundary conditions must provide. The matrices $\mathbf{P}$ and $\mathbf{Q}$ that transform the system to its generalized characteristic form may be computed stably only when the degeneracy of the system is zero; when the degeneracy is nonzero, stable similarity transforms exist that take both $\mathbf{A}$ and $\mathbf{B}$ to upper triangular matrices [21]. While not the characteristic form of the system, this *generalized upper triangular form* may be used in the same manner as the characteristic form for a more detailed boundary condition analysis.

Consider now a linear system in generalized upper triangular form (the generalized characteristic form may be used instead if available).

$$\mathbf{PAQv}_t + \mathbf{PBQv}_x = -\mathbf{PCQv} + \mathbf{Pf}(t, x) \tag{5.1}$$

Let $\rho_i = (\mathbf{PAQ})_{ii}$ and $\tau_i = (\mathbf{PBQ})_{ii}$. Because the coefficient matrix pencil is assumed regular, it is not possible for $\rho_i = \tau_i = 0$, and thus an output set assignment of v_i to equation i is implied. Given this output set assignment, each dependent variable is given as the solution to a (possibly degenerate) one-way wave.

A dynamic simulation implies advancing a solution forward in t. The values of the dependent variables v_i for which the associated characteristic direction ρ_i/τ_i is nonpositive are determined at $x = a$ by the outward-directed characteristics. Similarly, values associated with characteristics that have speeds greater than or equal to 0 are determined at $x = b$. Once the value of a dependent variable associated with an infinite speed characteristic is specified at one domain boundary, it is determined at the other as well. Initial conditions on $t = 0$ determine the variables associated with characteristics of speed 0 on the boundaries at all later times.

Let $\mathbf{v}_p$, $\mathbf{v}_r$, $\mathbf{v}_l$, and $\mathbf{v}_d$ be the variables associated with infinite, positive, negative, and zero speed characteristics respectively. The values of the dependent variables that are determined by characteristics at each boundary may be written as the solution to a system of the following form.

$$\begin{bmatrix} \mathbf{I}_p & 0 & 0 & 0 & \mathbf{I}_p & 0 & 0 & 0 \\ 0 & \mathbf{I}_l & 0 & 0 & & & & \\ 0 & 0 & 0 & \mathbf{I}_d & & & & \\ & & & & 0 & 0 & \mathbf{I}_r & 0 \\ & & & & 0 & 0 & 0 & \mathbf{I}_d \end{bmatrix} \begin{bmatrix} \mathbf{v}_p(a,t) \\ \mathbf{v}_l(a,t) \\ \mathbf{v}_r(a,t) \\ \mathbf{v}_d(a,t) \\ \mathbf{v}_p(b,t) \\ \mathbf{v}_l(b,t) \\ \mathbf{v}_r(b,t) \\ \mathbf{v}_d(b,t) \end{bmatrix} = \mathbf{g}(t,x) \tag{5.2}$$

This system represents the parts of the solution that are fully determined at $x = a$ and $x = b$ by characteristics; it is not in general possible to give the righthand side analytically. It can, however, be used to evaluate the information contained in the boundary conditions specified by the engineer. Each dependent variable v_i in the generalized upper triangular form is a linear combination of the original variables $\mathbf{u}$. Transforming back to these original variables, the system becomes

$$\begin{bmatrix} \mathbf{C}_p & \mathbf{C}_p \\ \mathbf{C}_l & \\ \mathbf{C}_d & \\ & \mathbf{C}_r \\ & \mathbf{C}_d \end{bmatrix} \begin{bmatrix} \mathbf{u}(a,t) \\ \mathbf{u}(b,t) \end{bmatrix} = \mathbf{f}(t,x) \tag{5.3}$$

Suppose the boundary conditions to be enforced at $x = a$ are given by $\mathbf{G}_a\mathbf{u} = h_1(t)$, and at $x = b$ by $\mathbf{G}_b\mathbf{u} = h_2(t)$. The boundary conditions determine a unique solution if

$$\begin{vmatrix} \mathbf{C}_p & \mathbf{C}_p \\ \mathbf{C}_l & \\ \mathbf{C}_d & \\ & \mathbf{C}_r \\ & \mathbf{C}_d \\ \mathbf{G}_a & \\ & \mathbf{G}_b \end{vmatrix} \neq 0 \tag{5.4}$$

If the boundary conditions are Dirichlet conditions, then $\mathbf{G}_a$ and $\mathbf{G}_b$ are real matrices, and this determinant may be evaluated numerically. For Neumann and Robin conditions, the coefficient matrix for the boundary conditions is operator valued, which makes evaluation of the determinant a symbolic calculation.

Finally, consider systems with a singular coefficient matrix pencil. The cost of verifying the conditions given by theorem 4.5.8 and corollary 4.5.9 under which differentiation of algebraic equations with respect to t is guaranteed to produce a regular coefficient matrix pencil is greater than the cost of simply performing the necessary differentiations. After differentiation, the generalized upper triangular form will reveal whether or not the differentiations produced a regular pencil.

This analysis and implementation may be summarized as follows.

1. Use Pantelides' algorithm to obtain an estimate of the index of the system with respect to both t and x. In the absence of numerical singularities of the relevant matrices, the algorithm will return the true indices.

2. Use the information returned by Pantelides' algorithm with the method of dummy derivatives to produce two reformulated systems that are low index with respect to t and with respect to x.

3. Differentiate any algebraic equations once with respect to t. Calculate the generalized eigenvalues of this new coefficient matrix pair. Calculate the matrices $\mathbf{P}$ and $\mathbf{Q}$ that transform the coefficient matrix pair to either its canonical form or its generalized upper triangular form.

The results of the above three calculations provides a great deal of information regarding the index and well-posedness of a particular unit model. In the absence of numerical singularities, Pantelides' algorithm returns the index of the system with respect to t directly. If $\nu_t \geq 2$, any reasonable method of lines semidiscretization in t will produce a high index DAE.

Well-posedness information based on the results of these three calculations may be summarized as follows.

1. If Pantelides' algorithm terminates because it is unable to generate a transversal, a unique solution does not exist and the problem is ill-posed.

2. If the number of initial conditions is less than to the number of t-differential variables in the t-dummy reformulation, the solution is not unique, and the problem is ill-posed. If the number of initial conditions is greater than to the number of t-differential variables in the t-dummy reformulation, the problem is overdetermined. It may be redundant or inconsistent; in the latter case no solution exists and the problem is ill-posed.

3. If the total number of boundary conditions is less than to the number of x-differential variables in the x-dummy reformulation, the solution is not unique, and the problem is ill-posed. If the number of boundary conditions is greater than the number of x-differential variables in the x-dummy reformulation, the problem is overdetermined. It may be redundant or inconsistent; in the latter case no solution exists and the problem is ill-posed.

4. If the number of boundary conditions at $x = a$ is less than the number of positive generalized eigenvalues, or the number of boundary conditions at $x = b$ is less than the number of negative generalized eigenvalues, the solution is not unique, and the problem is ill-posed.

5. If any generalized eigenvalues of the coefficient matrix pair are complex, the solution does not depend continuously on its data, and the problem is ill-posed.

6. If $\mathbf{C} = \mathbf{0}$ and any generalized eigenvalues of the coefficient matrix pair are given by $0/0$, the system fails the regularity condition numerically, so the solution is not unique and the problem is ill-posed.

7. If any eigenvalue given by τ/ρ, $\rho \neq 0$ has degeneracy 1, and $\nu_x < 2$, the solution does not depend continuously on its data, and the problem is ill-posed.

Again note that this analysis applies rigorously only to linear systems. Extensions based on local values may be made to semilinear and quasilinear systems, but very few general statements may be made about truly nonlinear distributed unit models.

5.2 Examples

5.2.1 Larry's problem: pressure-swing adsorption

Could the analyses outlined in the previous section enable a simulator to provide some insight into the cause of Larry's difficulties with the pressure-swing adsorption simulation? The first step is estimation of the index. Pantelides' algorithm differentiates the isotherm once before terminating, indicating that the index of the system with respect to t is 2, and thus immediately pointing to the underlying cause of the simulation failure. The original system had a high index with respect to t, which was preserved by the method-of-lines semidiscretization in t to produce a high index DAE.

The simulator could provide an equivalent dummy reformulation of the original PDE that has index 1 with respect to t. There are two possible dummy reformulations;

one is

$$
\begin{gathered}
\frac{\rho_B RT}{P}\sum_{i=1}^{3} q_i' + \frac{\epsilon}{P}P_t + u_z = 0 \\
\epsilon y_{i_t} + \frac{\rho_B RT}{P} q_i' + \frac{\epsilon y_i}{P} P_t + (uy_i)_z = 0, \quad i = 1 \dots 3 \\
\sum_{i=1}^{4} y_i = 1 \\
q_i - \frac{q_i^{sat} B_i (y_i P)^{\frac{1}{n_i}}}{1 + \sum_{j=1}^{3} B_j (y_j P)^{\frac{1}{n_j}}} = 0, \quad i = 1 \dots 3 \\
\left(1 + \sum_{j=1}^{3} B_j (y_j P)^{\frac{1}{n_j}}\right) q_i' \;\; + \\
\left(q_i \sum_{j=1}^{3} B_j P^{\frac{1}{n_j}} - q_i^{sat} B_i P^{\frac{1}{n_i}}\right)\left(\frac{1}{n_i}\right) y_i^{\left(\frac{1}{n_i - 1}\right)} y_{i_t} = 0, \quad i = 1 \dots 3
\end{gathered}
\tag{5.5}
$$

By item 3 in the analysis of the equations, only three initial conditions should be enforced.

Discretizing this system using the same upwind finite difference scheme and employing the same BDF integrator in time produces a low index DAE. Once the (redundant) initial conditions on $q_{i=1\dots3}$ are eliminated, the solution proceeds normally. Results for the first few operating cycles appear in figure 5-1.

In this case automated model analysis is able to immediately identify the root cause of the simulation failure. Furthermore, a simulator would be able to correct the underlying problem automatically, with no intervention on Larry's part.

5.2.2 Moe's problem: compressible flow

What about Moe's difficulties with his compressible flow simulation? Can a process simulator use these tools to help get the simulation working?

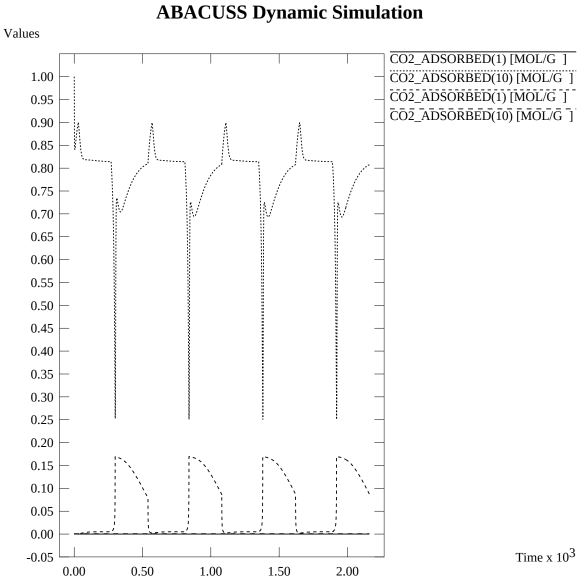

Figure 5-1: Simulation results for reformulated problem

In quasilinear form, the model equations are

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ h & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ u \\ p \\ h \\ i \end{bmatrix}_t + \begin{bmatrix} u & \rho & 0 & 0 & 0 \\ u^2 & 2\rho u & 1 & 0 & 0 \\ uh & \rho h - p & -u & \rho u & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ u \\ p \\ h \\ i \end{bmatrix}_x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ p - (\gamma - 1)\rho i \\ i - h + \frac{1}{2}u^2 \end{bmatrix} \tag{5.6}$$

Pantelides' algorithm, applied to determine the index with respect to t, locates no minimally structural subsets of equations. The index with respect to t is in fact 1. No dummy reformulation is necessary, and the number of dynamic degrees of freedom on $t = 0$ is three. Also, the system does not fail to meet the regularity condition based on structural criteria.

The coefficient matrices do not form a regular pencil. Because $\nu_t = 1$, $E^{(At)} \cap E^{(Dx)} = \phi$, and both $J(E^{(Dt)}, V^{(Dt)})$ and $J(E^{(At)}, V^{(At)})$ have full row rank for all physical values of ρ, theorem 4.5.8 guarantees that differentiating $E^{(At)}$ once with respect to t will produce a system with a regular coefficient matrix pencil.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ h & 0 & 0 & \rho & 0 \\ (1-\gamma)\,i & 0 & 1 & 0 & (1-\gamma)\,\rho \\ 0 & u & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \rho \\ u \\ p \\ h \\ i \end{bmatrix}_t + \begin{bmatrix} u & \rho & 0 & 0 & 0 \\ u^2 & 2\rho u & 1 & 0 & 0 \\ uh & \rho h + p & u & \rho u & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ u \\ p \\ h \\ i \end{bmatrix}_x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.7}$$

The system is quasilinear, so the coefficient matrices must be frozen at a point of interest. Consider the domain boundary at $x = 10$, and let conditions at $x = 10$ be $\rho = 79.6\ \frac{\mathrm{kg}}{\mathrm{m}^3}$, $u = 0.00\ \frac{\mathrm{m}}{\mathrm{s}}$, $p = 2.76\ \mathrm{MPa}$, $h = 86.6\ \mathrm{kJ}$, and $i = 86.6\ \mathrm{kJ}$. The frozen coefficient matrices are submitted to an eigensolver, such as the `LAPACK` routine `dgegv`. The result is three characteristic directions parallel to the t coordinate axis and two complex characteristic directions. The system is thus ill-posed in a neighborhood of these nominal values, and cannot be solved by a simulator as part of a dynamic simulation.

A process simulator could thus advise Moe that the equations, as he has entered them, are ill-posed. On review of the input, the sign error made in the energy balance (1.13) should be corrected.

$$(\rho h)_t + (\rho u h + u p)_x = 0 \tag{5.8}$$

The analysis may then be repeated for the corrected system. Now, the degeneracy is found to be zero, with two characteristic directions parallel to the t coordinate axis and three with slopes −170.32, 50.00, and 270.32 m/s in the (t, x) plane. The corrected problem is therefore well-posed

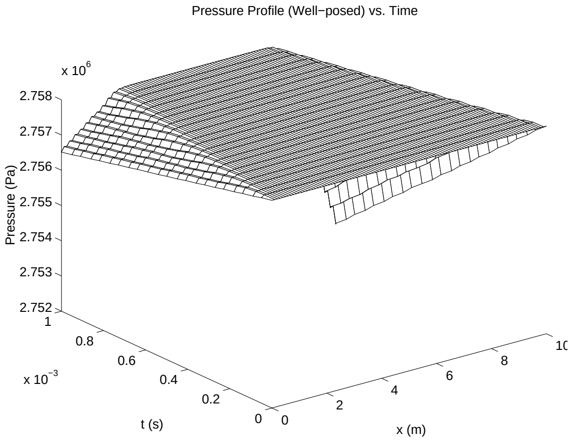

Figure 5-2: Corrected pipe pressure profile

Simulation results for the corrected problem appear in figure 5-2. As expected, a rarefaction enters the pipe from both ends. This time, the simulation failure was the result of a simple sign error on Moe's part. This sign error produced a strongly ill-posed system, which can be detected by a process simulator through the use of the analyses developed in this thesis.

5.2.3 Curly's problem: electric power transmission

Could the automatable analyses developed in this thesis help uncover the cause of Curly's electric power line simulation failure? The index of the system with respect to both t and x is zero; Pantelides' algorithm would correctly return no differentiations. Therefore, no reformulation is necessary. The coefficient matrices are linear and have two generalized eigenvalues $\pm 182,879$. The corresponding transformation matrices $\mathbf{P}$ and $\mathbf{Q}$ are

$$\mathbf{P} = \begin{bmatrix} -1.19E-3 & 1.00 \\ 1.19E-3 & 1.00 \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} -4.21E+2 & 4.21E+2 \\ 5.00E-1 & 5.00E-1 \end{bmatrix} \tag{5.9}$$

The canonical form of the system is

$$\begin{bmatrix} -5.47E-6 & \\ & 5.47E-6 \end{bmatrix} \mathbf{v}_t + \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \mathbf{v}_x + \begin{bmatrix} -1.40E+4 & 1.40E+4 \\ -1.40E+4 & 1.40E+4 \end{bmatrix} \mathbf{v} = \mathbf{0} \tag{5.10}$$

Several things are apparent from the canonical form. First, one boundary condition must be enforced at each end of the domain. The problem, as Curly has defined it, is ill-posed because the two boundary conditions enforced at the substation do not determine a unique solution. In this case, it means that he must obtain data from another substation at the other end of the line, in order to provide the required boundary condition at that end of the domain.

Also, once these measurements have been taken, the characteristic speeds give a time step size restriction. For a finite difference scheme, the time step must be limited by a CFL condition [81]. Here, that restriction is $\Delta t \leq \Delta x/182,879$.

Why, then, did the simplified model work so well? Analysis of the simplified model shows that the index with respect to t is 2. No initial conditions may be arbitrarily specified. Initializing u at an inconsistent value caused the small initial jump in current shown in the simulation results. So, there was in fact a problem with the simplified model, but it was less serious than the outright failure that befell the simulation based on the full model. Also, the canonical form of the simplified system consists of a single degenerate parabolic block with simple forcing. Two boundary conditions at the same domain endpoint therefore do determine a unique solution of the simplified model. Finally, there is no CFL condition limiting the time step.

In this case, the mathematical properties of the simplified model are very different from those of the full model. The analyses developed in this thesis uncover these differences, and may be used to provide very understandable feedback to the engineer; specifically, that he needs to remove a boundary condition at the left domain endpoint, and enforce one at the right. This means, for Curly, going out into the field and obtaining a new set of measurements at a new location, or inferring new information from existing data.

5.2.4 Shemp's problem: combustion kinetics

What about Shemp's difficulties with his combustion kinetics model? Can the tools developed in this thesis help diagnose the cause of the problem?

Pantelides' algorithm, applied to determine the index with respect to z, differentiates the third and seventh through eleventh equations a single time before terminating, thereby indicating that the index with respect to z is 2.

A dummy reformulation of the problem consists of the following modifications to the original equations

$$\begin{aligned} c_{i_t} + u' c_{i_z} + (c_i v_i)_z &= \omega_i \\ v_i &= -\frac{D_i}{x_i} x_i' \end{aligned} \tag{5.11}$$

together with the following new equations

$$\begin{aligned} \rho' x_i + \rho x_i' &= c_{i_z} \\ R\rho' T + R\rho T' &= 0 \\ T' &= \frac{\partial}{\partial z} g(z) \\ \rho_m' &= \rho' w_{mean} + \rho w_{mean}' \\ w_{mean}' \sum_{j=1}^{4} \frac{y_j}{w_j} + w_{mean} \sum_{j=1}^{4} \frac{y_j'}{w_j} &= 0 \\ u' \rho_m + u \rho_m' &= 0 \\ y_i' \rho_m + y_i \rho_m' &= w_i c_{i_z} \end{aligned} \tag{5.12}$$

However, after removing the conditions on $u(0)$ and $x_i(0)$, the simulation still fails during the initialization calculation. With c_{i_t} set to zero, the system is a DAE in z, so the characteristic analysis offers no further insight.

So what is wrong? It turns out that there is a problem with this particular model formulation. Consider equations 6 through 10 in the original model 1.18. Equation 9 must be assigned to T, because no other dependent variables appear in it. Equation 8 must then be solved for ρ, because only ρ and T appear in it, and equation 9 is solved for T.

Equations 6, 7, and 10 must then be assigned to some combination of $\mathbf{c}$, $\mathbf{y}$, ρ_m, and w_{mean}. However, they cannot be used to solve for $\mathbf{y}$, ρ_m, and w_{mean}. To see this, first use equation 7 to eliminate ρ_m, and rearrange the terms in the remaining equations to

$$\begin{aligned} w_{mean}\frac{y_i}{w_i} &= \frac{c_i}{\rho} \\ w_{mean}\sum\frac{y_i}{w_i} &= 1 \end{aligned} \tag{5.13}$$

and then examine the Jacobian of these equations with respect to w_{mean} and $\mathbf{y}$.

$$\begin{bmatrix} \frac{y_1}{w_1} & \frac{w_{mean}}{w_1} & 0 & 0 & 0 \\ \frac{y_2}{w_2} & 0 & \frac{w_{mean}}{w_2} & 0 & 0 \\ \frac{y_3}{w_3} & 0 & 0 & \frac{w_{mean}}{w_3} & 0 \\ \frac{y_4}{w_4} & 0 & 0 & 0 & \frac{w_{mean}}{w_4} \\ \sum\frac{y_i}{w_i} & \frac{w_{mean}}{w_1} & \frac{w_{mean}}{w_2} & \frac{w_{mean}}{w_3} & \frac{w_{mean}}{w_4} \end{bmatrix} \tag{5.14}$$

Clearly the Jacobian is singular, so the equations cannot be solved for ρ_m, $\mathbf{y}$, and w_{mean}.

Therefore, at least one of these equations must be solved for a concentration c_i. Because all c_i are differential variables, one of the differential equations involving c_{i_z} must then be assigned to an algebraic variable. Differentiating that equation once with respect to z then gives the derivative of that variable with respect to z as a function of $c_{i_{zz}}$; the equation assigned to c_i must be differentiated twice to eliminate it, and therefore the dummy reformulation is still high index.

What if a dummy reformulation in which c_i was an algebraic variable (i.e. a new dummy derivative c' had been introduced) was chosen instead of the one shown above? In this case, the differential variables would be $\mathbf{v}$ and $\mathbf{x}$. However, not all of the elements of $\mathbf{x}$ may be set independently; there is an implicit constraint that relates the x_i.

To see this constraint, one can rearrange the terms of equation 6 to $y_i/w_i = c_i/\rho_m$ and then sum to obtain

$$\sum\frac{y_i}{w_i} = \frac{1}{\rho_m}\sum c_i \tag{5.15}$$

Similarly, rearrange equation 3 to $\rho x_i = c_i$ and sum to produce

$$\rho\sum x_i = \sum c_i \tag{5.16}$$

Using this result to eliminate $\sum c_i$ from above gives

$$\sum \frac{y_i}{w_i} = \frac{\rho}{\rho_m} \sum x_i \tag{5.17}$$

Now, from equation 7,

$$\frac{\rho}{\rho_m} = \frac{1}{w_{mean}} \tag{5.18}$$

which may be used to eliminate ρ/ρ_m from above to obtain

$$\sum \frac{y_i}{w_i} = \frac{1}{w_{mean}} \sum x_i \tag{5.19}$$

Finally, inverting both sides and switching them produces

$$w_{mean} \frac{1}{\sum x_i} = \frac{1}{\sum \frac{y_i}{w_i}} \tag{5.20}$$

Any solution must satisfy the above relationship. It must also satisfy equation 10, which is

$$w_{mean} = \frac{1}{\sum \frac{y_i}{w_i}} \tag{5.21}$$

This is only possible if $\sum x_i = 1$, so not all x_i are independent.

This motivates the inclusion of the "correction factor" into the diffusion velocity equations found in some formulations [15]. In fact, if a new variable v_c is added to the righthand side of equation 2, and a new equation for the mass fractions y_i

$$\sum y_i = 1 \tag{5.22}$$

and their dummy derivatives

$$\sum y_i' = 0 \tag{5.23}$$

are appended to the system, the dummy reformulation becomes well-behaved. The fourth initial condition on y_i is no longer needed. Concentration profiles calculated for this formulation appear in figure 5-3.

This example, beyond being interesting in its own right, also highlights the fact that the automated analyses developed in this thesis are not capable of detecting and

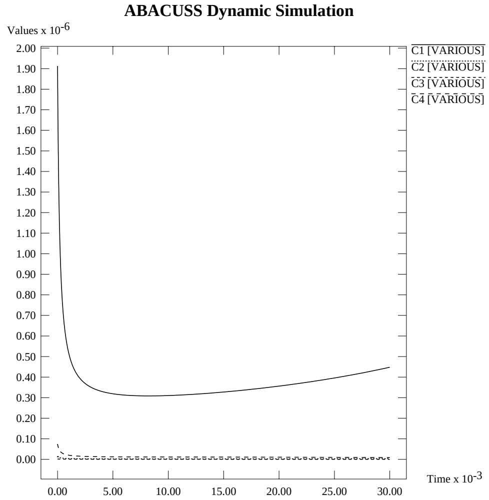

Figure 5-3: Concentration profiles for reformulated combustion model

dealing with every mathematical property of every possible distributed unit model. Here, a numerical singularity that goes undetected by structural analysis prevents the initialization calculation from succeeding, and manual reformulation of the model is required.

5.2.5 Moe's problem revisited: adaptive boundary conditions

The boundary condition evaluation method described earlier (5.4) may be modified slightly to create a method by which a simulator could automatically adapt boundary conditions as required to form a well-posed problem.

The Courant-Isaacson-Rees (CIR) scheme [19] solves hyperbolic partial differential equations using a linear finite difference approximation to the characteristic form of the model equations. Consider a quasilinear hyperbolic system in t and x over the domain $0 \leq x \leq 1$, $t \geq 0$.

$$\mathbf{u}_t + \mathbf{B}(\mathbf{u}, t, x)\mathbf{u}_x = \mathbf{f}(\mathbf{u}, t, x) \tag{5.24}$$

Let the domain be discretized into a set X of equispaced points, and let $x_i \in X$ be a particular point in that set. Initial data gives the values of the dependent variables $\mathbf{u}(x_i, 0)$.

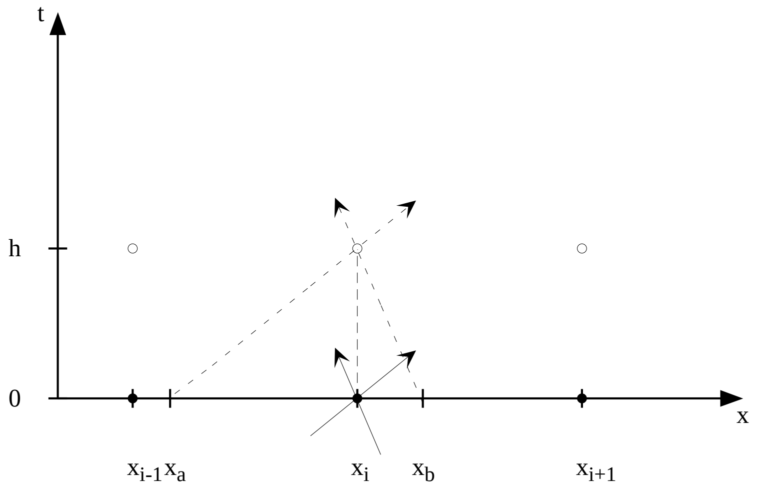

Figure 5-4: Stencil for CIR scheme

This scheme evaluates the coefficient matrix $\mathbf{B}$ at each node. For example, consider the i^{th} node in figure 5-4. The frozen coefficient system is

$$\mathbf{u}_t + \mathbf{B}(\mathbf{u}(0, x_i), 0, x_i)\mathbf{u}_x = \mathbf{f}(\mathbf{u}(0, x_i), 0, x_i) \tag{5.25}$$

Now, let $\mathbf{L}$ and $\mathbf{\Lambda}$ contain the left eigenvectors and the eigenvalues of the frozen coefficient matrix $\mathbf{B}(\mathbf{u}(0, x_i), 0, x_i)$, respectively, so the characteristic form of the frozen coefficient system is

$$\mathbf{L}\frac{d\mathbf{u}}{dt} = \mathbf{L}\mathbf{f}(\mathbf{u}(0, x_i), 0, x_i) \quad \text{along} \quad diag(\mathbf{I}dx) = diag(\mathbf{\Lambda}dt) \tag{5.26}$$

This system (5.26) is then used as an approximation to the system after a small increment h in time t. Using the explicit Euler finite difference approximation to the directional derivative along each characteristic gives equations of the form

$$\mathbf{l}_i \left(\frac{\mathbf{u}(h, x_i) - \mathbf{u}_i^*}{h} \right) = \mathbf{l}_i \mathbf{f}(\mathbf{u}(0, x_i), 0, x_i) \tag{5.27}$$

where $\mathbf{u}_i^*$ is the vector of values of $\mathbf{u}$ at the foot of the i^{th} characteristics of the frozen coefficient system, calculated by interpolation between values at grid points on $t = 0$. For example, in figure 5-4, $\mathbf{u}_a^* = \mathbf{u}(x_a, 0)$ is the value at the foot of characteristic a.

Let $v_i = \mathbf{l}_i\mathbf{u}_i^*$ and $g_i = \mathbf{l}_i f_i(\mathbf{u}_i^*, 0, x_i)$. Then the equations that give the value of $\mathbf{u}(x_i, h)$ are

$$\mathbf{L}\mathbf{u}(x_i, h) = \mathbf{v} + h\mathbf{g} \tag{5.28}$$

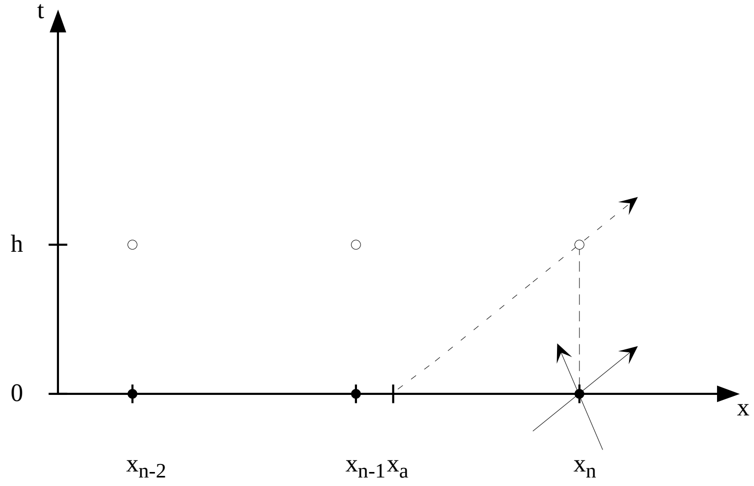

Figure 5-5: Modified CIR scheme for boundary point

This is the CIR scheme. For linear systems with simple or linear forcing, the coefficients on the left and righthand sides are constant, so calculating new values after a time step at each node only requires solving the same system with multiple righthand sides.

Performing the same approximation at a boundary node, but retaining only the outward-directed characteristics, produces the system that partially determines the solution at that boundary (5.4). If the characteristics associated with each line in that system is traced back from the next time $t+h$ to the current time t, and interpolation is used to determine the values at the feet of those characteristics, the righthand side is given in the same manner as in the CIR scheme, and is depicted graphically in figure 5-5.

Performing Gauss elimination with row and column pivoting on this (possibly underdetermined) system gives a number of pivot variables that are determined by the characteristic information. The simulator could take this information, together with the flowsheet topology and a specification of what variables refer to the same quantities in different unit models (for example, ρ in the pipe model refers to the same quantity as ρ_A in the model for valve 1), and attempt to set Dirichlet conditions on the remaining variables by equating values at the boundary to those in the adjacent unit, in order to form a fully determined system.

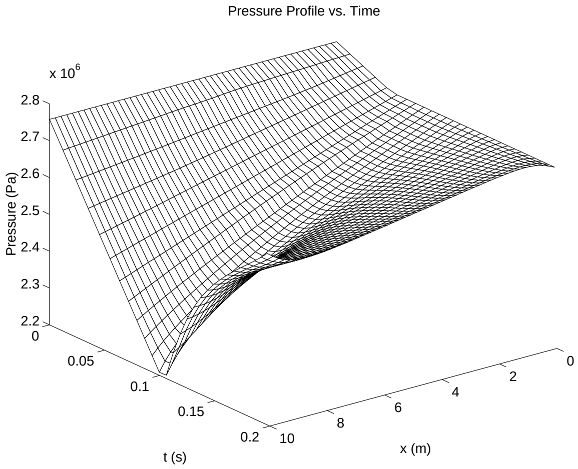

Figure 5-6: Pressure profile

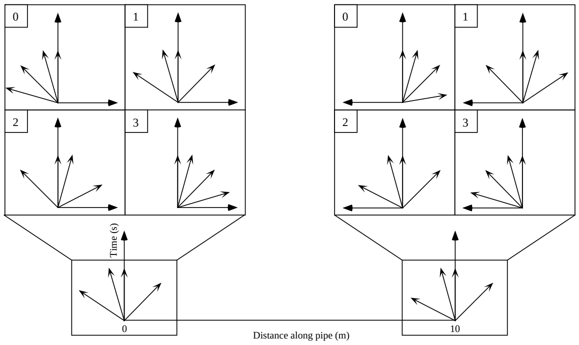

Figure 5-7: Characteristics and boundary condition requirements for Euler equations of compressible flow

For Moe's problem, consider use of this adaptive boundary condition scheme at the pipe ends, together with a Godunov scheme [32] using Roe's Riemann solver [72] on the domain interior. Using the LAPACK routine `rgg` to solve the generalized eigenvalue problem, and allowing the quantities that appear in both the pipe and the valve models to be u, ρ, p, and i, the method described is able to adapt the boundary conditions as needed to maintain a well-posed problem.

Possible characteristic directions at the endpoints of the domain and corresponding boundary condition regimes appear in figure 5-7. Three characteristics directed into the domain correspond to supersonic flow into the pipe at that end, and three boundary conditions are required. Two characteristics directed inward and one outward occurs when flow enters the pipe at subsonic conditions, and two boundary conditions are required. One characteristic directed inward corresponds to subsonic flow out of the pipe, which requires one boundary condition. Finally, no inward characteristics represents supersonic (or choked) flow out of the pipe, and no boundary conditions are required. The conditions at the two ends of the pipe may occur in any combination. Because it is based on the characteristics, the modified CIR scheme at the boundary together with the boundary condition selection method can correctly adapt to any combination of these flow regimes.

The pressure profile appears in figure 5-6. The dual rarefaction shown earlier in the short-time profile is replaced quickly by the evolving quasi-steady pressure gradient.

The boundary condition changes at the left end ($x = 0$) appear in figure 5-8. The short time results appear in the bottom frame, and results for the entire simulation appear in the upper frame. The method correctly adapts from one (ρ) to two (ρ and i) boundary conditions after the flow reversal. It correctly adjusts again when a sonic transition occurs, and enforces a third (p) boundary condition.

Boundary conditions changes enforced by the method at the right end ($x = 10$) appear in figure 5-9. No flow reversal occurs, and the method correctly enforces a single boundary condition on ρ until the sonic transition at approximately 0.1 seconds. The method removes this boundary condition when it is no longer needed, and obtains the solution at the boundary entirely from characteristic information after the sonic transition.

Without any intervention from Moe, or even any knowledge of the mathematical changes in the boundary condition requirements for well-posedness that occur at flow reversals and sonic transitions, a simulator employing this method could successfully adapt the boundary conditions. Moe needed only provide information regarding what variables refer to the same physical quantities in the different unit models.

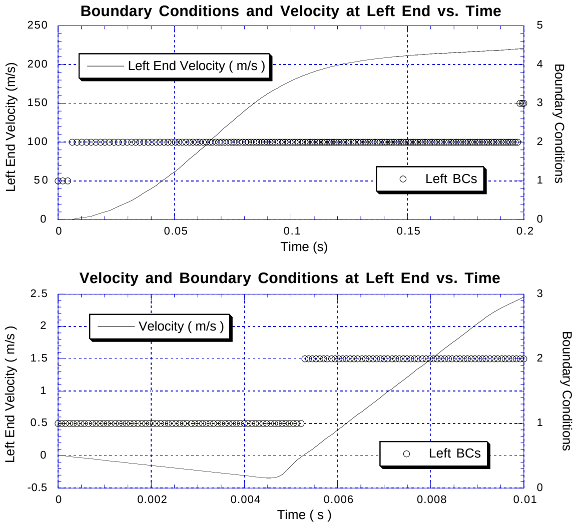

Figure 5-8: Velocity and Boundary Conditions at Left End of Pipe

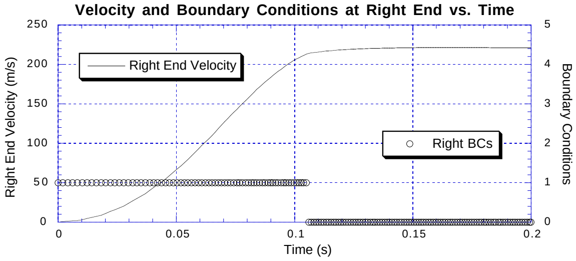

Figure 5-9: Velocity and Boundary Conditions at Right End of Pipe

Chapter 6

Conclusions and Discussion

6.1 Project summary

A generalization of the differentiation index of a DAE that applies to PDEs is presented. This generalized index is calculated with respect to a direction in the independent variable space of the equations. The index with respect to an arbitrary direction may be calculated by transforming the independent variables to a new coordinate system. Classical PDEs, such as the Navier-Stokes equations, as well as more general PDEs may have a high index with respect to one or more directions in the independent variable space.

This index analysis makes explicit all equations that must be satisfied by Cauchy data on a hyperplane orthogonal to the direction of interest. These equations may be simple algebraic, differential-algebraic, or partial differential-algebraic equations. In either of the latter two cases, additional data or side conditions may be required on subsurfaces of the original hyperplane. This index analysis may be used to determine restrictions on this additional data.

The most obvious application of this analysis is to consistent initialization problem for PDEs in a dynamic simulation. In this case the PDE is assumed to be an evolution problem. Marching solution techniques for such problems, whether built using the method of lines or the Rothe method, require consistent data on the initial hyperplane $x_j = x_{j0}$, where x_j is the evolution variable.

The index with respect to time also has applications in numerical solution using the method of lines, in the same way that the differentiation index of DAEs is used in construction of robust, automated integration methods that preserve all solution invariants [8, 25, 56]. One of the strengths of the differentiation index of a PDE as developed in this work is that it is a very natural generalization of the differentiation index of a DAE. As such, it allows algorithms like the Pantelides' algorithm [63] and the dummy derivative method [56], that are based on the DAE differentiation index, to be applied to PDEs as well, in a very straightforward manner.

This project also contributes a characteristic-like analysis of general first order systems. This analysis is applicable to a much broader class of equations than classical characteristic analysis, which applies only to strictly hyperbolic systems. This new characteristic-like analysis is built on a canonical form for first order systems, which is analogous to the characteristic form of a strictly hyperbolic system, but may be obtained for a much larger class of systems.

A property of the canonical form, the degeneracy, is defined. For partial differential equations, it is the degeneracy, rather than the index, that gives sufficient conditions on forcing function and data differentiability that guarantee existence of a smooth solution. The canonical form also provides requirements on location of boundary conditions that are necessary for existence and uniqueness of solutions.

The unrestricted solution to a first order system is proven to depend continuously on its data iff the generalized eigenvalues of the coefficient matrix pair are strictly real and of degeneracy zero. By proving this result, localization and linearization results used in the classical analysis of strictly hyperbolic systems are shown to be applicable to the broader class of systems considered in this thesis. Continuous dependence on data is also shown for the limited but important special class of restricted solutions that arise for equations with a diffusion term.

All of the information provided by this characteristic-like analysis comes from the generalized eigenvalues and eigenvectors of the coefficient matrix pair. These generalized eigenvalues and eigenvectors may be calculated using public domain code, which means that these analyses may also be performed automatically by a dynamic process simulator.

6.2 Future work

6.2.1 Improvements in the analyses

Several gaps in the analysis of even linear systems do exist. First, better identification of ill-posed linear problems with linear forcing as either weakly well-posed or strongly ill-posed would be useful. Because the former may often be solved successfully by a high order numerical method, and the latter may arise very easily via simple sign errors, a simulator should have the ability to distinguish between the two cases.

Second is reliance on structural algorithms as part of the index analysis. In practice for chemical engineering systems, these algorithms have proven very effective, but electric circuit simulations frequently produce systems with numerical singularities. Given a linear system, it may be possible to construct an algorithm that is numerical in nature and can handle singularities; however, such an algorithm might need to include symbolic operations or be significantly redesigned in order to deal with linear forcing.

Also, a better analysis of restricted solutions is needed. For example, the generalized characteristic analysis developed in this project can determine that a boundary condition must be given at each end of a one dimensional domain for both the forward and backward heat equations, and that a restricted solution must be used. It cannot then say which restricted solution is strongly ill-posed, and which is well-posed. It can only determine that the unrestricted solution determined by specification of two boundary conditions at the same domain endpoint is strongly ill-posed.

Consideration of semilinear and quasilinear systems requires additional attention. While analogy with known results for linear time varying DAEs and strictly hyperbolic and parabolic systems provides insight into how the analysis of linear systems might be expected to change upon consideration of quasilinear systems, formal consideration of such problems must still be performed.

6.2.2 The relationship between discretization and index

Also, there is the question of whether or not the index of a given method of lines discretization is equal to the index of the original PDE with respect to the evolution variable. Let the j^{th} independent variable be the evolution variable. In most method of lines discretization schemes, each of the dependent variables u_i on a (typically bounded) surface of the form $x_j = c$ is described by a finite set of parameters $\tilde{\mathbf{u}}_i$, such as the values of u at a set of nodes or the coefficients of a series. Let P be the set of all interior partial differential operators on surfaces of the form $x_j = c$, and let $\tilde{P}$ be the set of all real-valued matrices. Then, define a method-of-lines discretization scheme g as a function that maps $P \rightarrow \tilde{P}$. An interior partial differential operator $\mathcal{L}_k \in P$ is then represented by a relation $g(\mathcal{L}_k) = \mathbf{L}_k \in \tilde{P}$ between parameters; $\mathcal{L}_k u_i$ is approximated by $\mathbf{L}_k \tilde{\mathbf{u}}_i$.

When all interior partial differential operators and dependent variables have been represented in this manner, the result is a DAE in the evolution variable x_j. Values of or relationships between subsets of these discretized variables are then specified in order to enforce the boundary conditions. The resulting DAE is the discretization. The solution is typically advanced in the evolution variable using a numerical DAE solver.

Definition 6.2.1 *An x_j method of lines discretization of a PDE is a DAE in x_j that approximates the solution of that PDE.*

Definition 6.2.2 *An x_j method of lines discretization is called index-preserving iff its differentiation index is equal to the index with respect to x_j of the original PDE.*

It is difficult to make any broad *a priori* statements about which discretizations are index-preserving. Consider a linear, two dimensional, first order system of the following form, which will be solved numerically using an x_1 method of lines discretization, with the goal of identifying conditions on the discretization under which index preservation may be guaranteed.

$$\mathbf{A}\mathbf{u}_{x_1} + \mathbf{B}\mathbf{u} = \mathbf{f}(x_1, x_2) \tag{6.1}$$

Here $\mathbf{A} \in \mathbb{R}^{n \times n} \subset P^{n \times n}$, $\mathbf{B} \in P^{n \times n}$, where $P = \{\mathcal{L} | \mathcal{L} = \sum_\tau c_\tau (\frac{\partial}{\partial x_2})^\tau, c_\tau \in \mathbb{R}, \tau \in \mathbb{N}\}$. Observe that, in the case of a first order system, $\tau \geq 2 \Rightarrow c_\tau = 0$. Also recall that $\langle P, +, \times \rangle$ and thus $\langle P^{n \times n}, +, \times \rangle$ are both rings [54].

Now, consider an x_1 method of lines discretization of the system, where each dependent variable u_i is represented by a set of k parameters $\tilde{\mathbf{u}}_i$, and the partial derivative $\frac{\partial}{\partial x_2}$ is represented by the relationship between parameters given by the matrix $\mathbf{D}$; in other words, $\frac{\partial u_i}{\partial x_2}$ is approximated in the discrete system by $\mathbf{D}\mathbf{u}_i$. The discretized system is then

$$\tilde{\mathbf{A}} \tilde{\mathbf{u}}_{x_1} + \tilde{\mathbf{B}} \tilde{\mathbf{u}} = \tilde{\mathbf{f}}(x_1, x_2) \tag{6.2}$$

where $\tilde{\mathbf{A}}, \tilde{\mathbf{B}} \in \tilde{P}^{n \times n}$. Here $\tilde{P} = \{\mathbf{L} | \mathbf{L} = \sum_\tau c_\tau \mathbf{D}^\tau, c_\tau \in \mathbb{R}, \tau \in \mathbb{N}, \mathbf{D} \in \mathbb{R}^{k \times k}\}$. The discretization maps each continuous interior partial differential operator $c_\tau(\frac{\partial}{\partial x_2})^\tau$ to a discrete operator $c_\tau \mathbf{D}^\tau$. Examples of such discretizations include finite differences, finite elements, and spectral approximations on a single grid.

Theorem 6.2.3 *If $\mathbf{D}$ is invertible and $\mathbf{D}^\nu = \mathbf{D}^\tau \Leftrightarrow \nu = \tau$, then P and $\tilde{P}$ are isomorphic.*

Proof. Noting that addition is defined on $\tilde{P}$ as simply

$$a + b = \sum_\tau a_\tau \mathbf{D}^\tau + \sum_\tau b_\tau \mathbf{D}^\tau = \sum_\tau (a_\tau + b_\tau) \mathbf{D}^\tau$$

it is easy to verify that $\left\langle \tilde{P}, + \right\rangle$ is an abelian group. Defining multiplication in the usual way,

$$a \times b = \left(\sum_\tau a_\tau \mathbf{D}^\tau \right) \left(\sum_\nu b_\nu \mathbf{D}^\nu \right) = \sum_\tau \sum_\nu a_\tau b_\nu \mathbf{D}^{\tau + \nu}$$

clearly multiplication is associative, and both left and right distributive over addition. Therefore $\left\langle \tilde{P}, +, \times \right\rangle$ is a ring.

Let the discretization $\phi : P \to \tilde{P}$ be the mapping defined by

$$\phi \left(\mathcal{L} = \sum_\tau c_\tau \left(\frac{\partial}{\partial x} \right)^\tau \right) = \sum_\tau c_\tau \mathbf{D}^\tau$$

Clearly $\phi(a+b) = \phi(a) + \phi(b)$ and $\phi(a \times b) = \phi(a) \times \phi(b)$, so $\phi : P \to \tilde{P}$ is a homomorphism. Furthermore, because $\mathbf{D}$ is invertible and $\mathbf{D}^{\nu} = \mathbf{D}^{\tau}$ iff $\nu = \tau$, ϕ is one-to-one and onto. Thus P and $\tilde{P}$ are isomorphic. □

If P and $\tilde{P}$ are isomorphic and $\mathcal{A}_j$ is 1-full according to the definition given in [54], then there will exist a set of row operations $\tilde{\mathcal{R}}$ in $\tilde{P}$ such that $\tilde{\mathcal{A}}_j$ has the form

$$\tilde{\mathcal{R}}\tilde{\mathcal{A}}_j = \begin{bmatrix} \tilde{\mathbf{D}} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{H}} \end{bmatrix} \tag{6.3}$$

where $\tilde{\mathbf{D}}$ is a diagonal matrix with nonzero entries on the diagonal.

However, even with isomorphism, it is impossible to guarantee that every member of $\tilde{P}$ is invertible. Thus, 1-fullness of the j^{th} derivative array equations is not necessarily equivalent to 1-fullness of the discretization. It only implies that the j^{th} derivative array equations of the discretization may be transformed by a series of row operations to a system with the structure given above (6.3); the individual blocks may or may not be invertible. If a diagonal element of $\tilde{D}$ is a singular matrix, the j^{th} derivative array equations for the discretization will not be 1-full, and the discretization will not be index preserving.

The problem is compounded when more than one interior direction is considered. The difficulty lies in the fact that the operator-valued original system is fundamentally different from the block matrix-valued discretized system. In the original system, operators commute over multiplication but do not possess multiplicative inverses. In the discretized system, block matrices do not in general commute, but matrices may possess multiplicative inverses. In the one-dimensional case, with a single discrete representation of the interior partial differential operator, isomorphism with the original system is possible because every matrix commutes over multiplication with itself. Due to the aforementioned differences in the algebraic structures of more general PDEs and discretizations of PDEs, *a priori* guarantees of index preserving properties of a particular scheme will be extremely difficult to prove. A structural analysis may be more tractable; for example, it may be possible to show that a discretization g that maps every hard zero to a zero block and every indeterminate entry to a square block with a transversal preserves the index in some structural sense.

The method chosen to enforce boundary conditions can influence whether or not a given discretization preserves the index. Two different methods of enforcing the same boundary conditions, used with the same scheme, can produce discretizations of differing index. Also, two different schemes used with the same method for enforcing boundary conditions can also produce discretizations of differing index.

Example 13 *Consider the heat equation, with evolution variable x_1.*

$$u_{x_1} - u_{x_2x_2} = 0 \tag{6.4}$$

on the domain $0 \leq x_1, 0 \leq x_2 \leq 1$, with u given at the boundaries by $u(0, x_1) = f_1(x_1)$ and $u(1, x_1) = f_2(x_1)$. The index of the model equation with respect to x_1 is zero.

Now, discretize the system by the Galerkin finite element method with linear elements. For K elements and $K+1$ nodes, the x_1 method of lines discretization is

$$\Delta x_2 \cdot \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & & & \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{6} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{6} & \frac{1}{3} & \frac{1}{6} \\ & & & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix}_{x_1} + \tag{6.5}$$

$$\frac{1}{\Delta x_2} \cdot \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix} = \mathbf{0}$$

Here $\mathbf{u} \in \mathbb{R}^{K+1}$ is the values of u at each of the $K+1$ nodes. The index of this system is zero, so the scheme is index-preserving. However, the system does not yet incorporate the boundary conditions, which may be implemented in one of several ways.

Consider first the penalty or "big number" approach. A suitably large number $\frac{1}{\epsilon}$ is added to the diagonal elements of the stiffness matrix that correspond to u_0 and u_K, and the product of that large number and either $f_1(t)$ or $f_2(t)$ is added to the *righthand side in the same row. If $\frac{1}{\epsilon}$ is much larger than the other elements of the system matrix, then $u_0 \rightarrow f_1(t)$ and $u_K \rightarrow f_2(t)$. The system becomes*

$$\Delta x_2 \cdot \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & & & \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{6} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{6} & \frac{1}{3} & \frac{1}{6} \\ & & & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix}_{x_1} + \frac{1}{\Delta x_2} \cdot \begin{bmatrix} 1+\frac{1}{\epsilon} & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1+\frac{1}{\epsilon} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix} = \begin{bmatrix} \frac{1}{\epsilon} f_1(t) \\ \mathbf{0} \\ \vdots \\ \frac{1}{\epsilon} f_2(t) \end{bmatrix} \tag{6.6}$$

This method of enforcing the boundary conditions, together with the linear Galerkin finite element formulation, does not alter the index of the system. The discretization is therefore index-preserving.

However, this approach in general worsens the condition number of the system to be solved during numerical integration. An alternative approach is to simply replace the finite element equations for u_0 and u_k with the boundary condition equations. Under such an approach, the discrete system is

$$\Delta x_2 \cdot \begin{bmatrix} 0 & 0 & & & \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{6} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{6} & \frac{1}{3} & \frac{1}{6} \\ & & & 0 & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix}_{x_1} + \frac{1}{\Delta x_2} \cdot \begin{bmatrix} 1 & & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix} = \begin{bmatrix} f_1(t) \\ 0 \\ \vdots \\ f_2(t) \end{bmatrix} \tag{6.7}$$

While this approach may be used to avoid conditioning problems, the index of the system has increased from zero to one. In fact, the system is special index-1. This discretization is not index-preserving.

Now, consider the same boundary condition implementation, but with a discrete system formulated using a lumped mass matrix. In one dimension, the mass matrix is lumped by simply summing off-diagonal elements and adding them to the diagonal. Again this is an index-preserving discretization; the system after enforcing boundary conditions as above is

$$\Delta x_2 \cdot \begin{bmatrix} 0 & & & & \\ & \frac{2}{3} & & & \\ & & \ddots & & \\ & & & \frac{2}{3} & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix}_{x_1} + \\ \frac{1}{\Delta x_2} \cdot \begin{bmatrix} 1 & & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix} = \begin{bmatrix} f_1(t) \\ 0 \\ \vdots \\ f_2(t) \end{bmatrix} \tag{6.8}$$

Under such a discretization, direct replacement of two finite element equations with boundary conditions does not alter the index of the discrete system, and the discretization is index-preserving.

Analysis of method of lines discretizations must therefore consider both the scheme and the boundary condition enforcement method together. The interaction of these two parts of a discretization can determine whether or not the overall discretization is index-preserving.

6.2.3 New network solution techniques

The generalized characteristic analysis developed in this thesis also provides information that may be used not only to analyze, but also to more effectively *solve* flowsheets that include distributed unit models. Two general areas are apparent: selection of an appropriate discretization scheme for each particular distributed model, and construction of new solution methods that are more efficient in a network context.

For linear systems[1], the characteristics are constant. One approach to using the characteristic analysis presented here would be as follows: for the hyperbolic part identified by the analysis, the generalized characteristic form may be used to derive a finite difference stencil that provides proper upwinding, in the same manner as the classical CIR scheme [19] for strictly hyperbolic systems, and then calculate the associated maximum time step. An implicit finite difference scheme may be derived for the algebraic, differential, and parabolic parts. More generally, one might simply assign variables associated with the parabolic, differential, or algebraic parts to the implicit part, and assign the remaining variables to the explicit part of a mixed implicit-explicit discretization.

Because the solution to weakly-well posed systems depends on derivatives of the forcing functions and data, this analysis could select a higher-order scheme that more accurately resolves interior partial derivatives whenever such a system is detected. For nondegenerate systems, a cheaper low order discretization might be more appropriate. Similarly, a system that consists of only a hyperbolic part might be best solved by an explicit scheme, while one that includes a parabolic or differential part might be best solved using an implicit scheme. This information could either be provided to the engineer to assist with generation of an appropriate discretization, or perhaps used by the simulator to select a discretization scheme. In either case, the choice would be motivated by the mathematical properties of the system itself.

For systems with a hyperbolic part, the CIR-like scheme employed in Moe's problem at the domain boundaries may be taken one step further. It might be possible to decouple the time steps taken by the BDF time integrator for adjacent lumped blocks from the time step taken by the discretization used in the distributed unit,

[1] Here the term "linear system" refers to constant coefficient partial differential equations; forcing functions and algebraic equations may be nonlinear, so long as the system is equivalent to the systems with singular coefficient matrix pencils considered in the previous chapter.

and employ waveform relaxation [41, 60, 74] to match the variables at the boundary. This would free the BDF integrator from the generally more restrictive time step requirements of the distributed unit discretization, and might therefore reduce the overall computational time required to perform a simulation.

For systems like the simplified telegrapher's equations that consist only of a degenerate parabolic block, this may be taken a step further. Because the solution of such a block does not depend on t, it is equivalent to an ODE in x. The BDF integrator for the remaining lumped units in the flowsheet can then take whatever time steps it needs to maintain error control, and simply calculate the solution on the transmission line at each step. In fact, a second BDF integrator may be used to start from boundary conditions at one end of the line, and advance them over the line to the other end. This has the effect of generating an adaptive grid automatically, as the integrator selects intervals ("steps" in x) as needed to maintain the error below a specified level.

Bibliography

[1] A. D. Aleksandrov, A. N. Kolmgorov, and M. A. Lavrent'ev, editors. *Mathematics: Its Content, Methods, and Meaning.* American Mathematical Society, Providence, Rhode Island, 1962.

[2] R. Allgor, M. Berrera, L. Evans, and P. I. Barton. Optimal batch process development. *Computers and Chemical Engineering*, 20(6/7):885–896, 1996.

[3] M. Andersson. *OMOLA - An Object-Oriented Language for Model Representation.* PhD thesis, Lund Institute of Technology, Lund, Sweden, 1990.

[4] R. Aris. *Vectors, Tensors, and the Basic Equations of Fluid Dynamics.* Prentice-Hall Inc., Englewood Cliffs, NJ, 1962.

[5] P. I. Barton and C. Pantelides. Modeling of combined discrete/continuous processes. *AIChE Journal*, 40(6):966–979, June 1994.

[6] M. Berzins, P. Dew, and R. Furzeland. Developing software for time-dependent problems using the method of lines and differential-algebraic integrators. *Applied Numerical Mathematics*, 5:375–397, 1989.

[7] R. B. Bird, W. E. Stewart, and E. E. Lightfoot. *Transport Phenomena.* John Wiley and Sons, New York, 1960.

[8] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* North Holland, New York, 1989.

[9] P. Bujakiewicz. *Maximum weighted matching for high index differential algebraic equations.* PhD thesis, Delft University of Technology, Delft, The Netherlands, 1994.

[10] S. L. Campbell. *Singular systems of differential equations.* Pitman Publishing, San Francisco, 1982.

[11] S. L. Campbell and C. W. Gear. The index of general nonlinear DAEs. *Numerische Mathematik*, 72:173–196, 1995.

[12] S. L. Campbell and W. Marszalek. The index of an infinite dimensional implicit system. *Mathematical Modelling of Systems*, 1997.

[13] F. E. Cellier. *Combined Continuous/Discrete System Simulation by use of Digital Computers: Techniques and Tools.* PhD thesis, Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, 1979.

[14] F. E. Cellier and A. E. Blitz. GASP-V: A universal simulation package. In *Proceedings of the 8th AICA Congress on Simulation of Systems*, Delft, The Netherlands, 1976.

[15] T. P. Coffe and J. M. Heimerl. A transport algorithm for premixed, laminar, steady-state flames. *Combustion and Flame*, 43, 1981.

[16] J. D. Cole and K. B. Yount. Applications of dynamic simulation to industrial control problems. In *Advances in Instrumentation and Control*, volume 48, pages 1337–1344, Chicago, USA, 1993.

[17] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87:171–200, 1990.

[18] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 2. Interscience Publishers, New York, 1962.

[19] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, V:243–255, 1952.

[20] S. R. Cvetkovic, A. P. Zhao, and M. Punjani. An implementation of the vectorial finite element analysis of anisotropic waveguides through a general-purpose PDE software. *IEEE Transactions on Microwave Theory and Techniques*, 42(8):1499–1505, 1994.

[21] J. Demmel and B. Kågström. The generalized Schur decomposition of an arbitrary pencil A - λB: Robust software with error bounds and applications. Part I: Theory and algorithms. *ACM Transactions on Mathematical Software*, 19(2):160–174, June 1993.

[22] C. R. Doering and J. D. Gibbon. *Applied Analysis of the Navier-Stokes Equations*. Cambridge University Press, Cambridge, 1995.

[23] I. S. Duff. On algorithms for obtaining a maximum transversal. *ACM Transactions on Mathematical Software*, 7(3):315–330, September 1981.

[24] H. Elmqvist. *A Structured Model Language for Large Continuous Systems*. PhD thesis, Lund Institute of Technology, Lund, Sweden, 1978.

[25] W. F. Feehery and P. I. Barton. A differentiation-based approach to dynamic simulation and optimization with high-index differential-algebraic equations. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation*. SIAM, 1996.

[26] W. F. Feehery and P. I. Barton. Dynamic simulation and optimization with inequality path constraints. *Computers and Chemical Engineering*, 20(S):S707–S712, 1996.

[27] W. F. Feehery and P. I. Barton. Dynamic optimization with state variable path constraints. *Computers and Chemical Engineering*, 22(9):1241–1256, 1998.

[28] W. F. Feehery, J. Tolsma, and P. I. Barton. Efficient sensitivity analysis of large-scale differential-algebraic systems. *Applied Numerical Mathematics*, 25(1):41–54, 1997.

[29] J. B. Fraleigh. *A First Course in Abstract Algebra*. Addison-Wesley Publishing Company, New York, fourth edition, 1989.

[30] F. R. Gantmacher. *The Theory of Matrices*. Chelsea Publishing Company, New York, 1959.

[31] C. W. Gear. Differential-algebraic equation index transformations. *SIAM Journal of Scientific and Statistical Computing*, 9(1):39–47, 1988.

[32] S. K. Godunov, A. V. Zabrodin, and G. P. Prokopov. A computational scheme for two-dimensional non stationary problems of gas dynamics and calculation of the flow from a shock wave approaching a stationary state. *USSR Computational Mathematics and Mathematical Physics*, 1:1187–1219, 1962.

[33] G. Golub and C. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[34] V. G. Grassi. Dynamic simulation as a tool to integrate process design and process control. In *Advances in Instrumentation and Control*, volume 48, pages 1345–1365, Chicago, USA, 1993.

[35] S. I. Grossman. *Calculus*. Academic Press, New York, 1981.

[36] M. Günther and Y. Wagner. Index concepts for linear mixed systems of differential-algebraic and hyperbolic-type equations. Submitted to *SIAM Journal on Scientific and Statistical Computing*, January 1999.

[37] E. Hairer, G. Wanner, and S. P. Nørsett. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, New York, second edition, 1993.

[38] J. O. Hirschfelder, C. F. Curtiss, and D. E. Campbell. The theory of flames and detonations. In *Proceedings of the Nineteenth Symposium (International) on Combustion*, Pittsburgh, PA, 1953.

[39] R. Jarvis. DASOLV: A differential-algebraic equation solver. Technical report, Imperial College, London, March 1992.

[40] A. Jeffrey. *Quasilinear hyperbolic systems and waves.* Pitman Publishing, London, 1976.

[41] J. A. Jelen. Multidomain direct method and its applicability to the local timestep concept. *Numerical Methods for Partial Differential Equations*, 10:85–101, 1994.

[42] P. Keast and P. Muir. EPDECOL: A more efficient PDECOL code. *ACM Transactions on Mathematical Software*, 17(2):153–166, June 1991.

[43] E. S. Kikkinides and R. T. Yang. Simultaneous SO_2/NO_x removal and SO_2 recovery from flue gas by pressure swing adsorption. *Industrial and Engineering Chemistry Research*, 30(8):1981–1989, 1991.

[44] H.-O. Kreiss and J. Lorenz. *Initial-Boundary Value Problems and the Navier-Stokes Equations.* Academic Press, New York, 1989.

[45] A. Kröner, W. Marquardt, and E. D. Gilles. Computing consistent initial conditions for differential-algebraic equations. *Computers and Chemical Engineering*, 16(S):131–138, 1992.

[46] H. P. Langtangen. *Computational Partial Differential Equations, Numerical Methods and Difpack Programming.* Springer-Verlag, New York, 1999.

[47] B. V. Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.

[48] H. M. Lieberstein. *Theory of Partial Differential Equations.* Academic Press, New York, 1972.

[49] R. Liska and M. Y. Shashkov. Algorithms for difference schemes construction on non-orthogonal logically rectangular meshes. In *Proceedings of International Symposium on Symbolic and Algebraic Computation ISSAC'91*, New York, 1991.

[50] R. Liska, M. Y. Shashkov, and A. V. Solovjov. Support-operators method for PDE discretization: symbolic algorithms and realization. *Mathematics and Computers in Simulation*, 35:173–183, 1994.

[51] E. J. Longwell. Dynamic modeling for process control and operability. In *Advances in Instrumentation and Control*, volume 48, pages 1323–1336, Chicago, USA, 1993.

[52] N. Madsen and R. F. Sincovec. PDECOL, general collocation software for partial differential equations. *ACM Transactions on Mathematical Software*, 5(3):326–351, September 1979.

[53] W. Marquardt. Trends in computer-aided process modeling. In *Proceedings of PSE '94*, 1994.

[54] W. S. Martinson and P. I. Barton. A differentiation index for partial differential-algebraic equations. In press: *SIAM Journal on Scientific Computing*, November 1998.

[55] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE.* McGraw-Hill, New York, 2nd edition, 1993.

[56] S. E. Mattsson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal on Scientific Computing*, 14(3):677–692, May 1993.

[57] C. Mayer, W. Marquardt, and E. D. Gilles. Reinitialization of DAEs after discontinuities. *Computers and Chemical Engineering*, 19(S):S507–S512, 1995.

[58] J. A. Miller, M. D. Smooke, R. M. Green, and R. J. Kee. Kinetic modeling of the oxidation of ammonia in flames. *Combustion Science and Technology*, 34:149–176, 1983.

[59] S. Mizohata. *The theory of partial differential equations*. Cambridge University Press, 1973.

[60] A. R. Newton. Techniques for the simulation of large-scale integrated circuits. *IEEE Transactions on Circuits and Systems*, 26(9):741–749, September 1986.

[61] M. Oh. *Modelling and Simulation of Combined Lumped and Distributed Processes*. PhD thesis, Imperial College of Science, Technology, and Medicine, London, 1995.

[62] M. Oh and C. C. Pantelides. Process modelling tools and their application to particulate processes. *Powder Technology*, 87:13–20, 1996.

[63] C. C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, 9(2):213–231, March 1988.

[64] T. Park and P. I. Barton. State event location in differential-algebraic models. *ACM Transactions on Modelling and Computer Simulation*, 6(2):137–165, 1996.

[65] T. Park and P. I. Barton. Implicit model checking of logic-based control systems. *AIChE Journal*, 43(9):2246–2260, 1997.

[66] J. D. Perkins and R. W. H. Sargent. SPEEDUP: A computer program for steady-state and dynamic simulation and design of chemical processes. *AIChE Symposium Series*, 78, 1982.

[67] N. Peters and J. Warnatz, editors. *Numerical Methods in Laminar Flame Propagation*, Wiesbaden, 1982. Friedr. Vieweg & Sohn.

[68] L. R. Petzold. A description of DASSL: a differential-algebraic equation solver. In *Proceedings of 10th IMACS World Congress*, Montreal, Canada, 1982.

[69] L. R. Petzold. Differential/Algebraic equations are not ODE's. *SIAM Journal on Scientific and Statistical Computing*, 3(3):367–384, September 1982.

[70] B. Pfeiffer and W. Marquardt. Symbolic semi-discretization of partial differential equation systems. In *IMACS Symposium SC-93*, Lille, France, 1993.

[71] G. Reißzig, W. S. Martinson, and P. I. Barton. Differential-algebraic equations of index 1 may have an arbitrarily high structural index. In press: *SIAM Journal on Scientific Computing*, October 1999.

[72] P. L. Roe. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics*, 18:337–365, 1986.

[73] R. Saurel, M. Larini, and J. C. Loraud. Exact and approximate Riemann solvers for real gases. *Journal of Computational Physics*, 112:126–137, 1994.

[74] A. R. Secchi, M. Morari, and E. C. Biscaia, Jr. The waveform relaxation method in the concurrent dynamic process simulation. *Computers in Chemical Engineering*, 17(7):683–704, 1993.

[75] D. Sédès. Modelling, simulation and process safety analysis. A case study: The formaldehyde process. Technical report, MIT, Cambridge, 1995.

[76] G. Sewell. *Analysis of a Finite Element Method: PDE/Protran*. Springer-Verlag, New York, 1985.

[77] W. Shönauer and E. Schnepf. FIDISOL: A 'black box' solver for partial differential equations. *Parallel Computing*, 6:185–193, 1988.

[78] R. F. Sincovec, A. M. Erisman, E. L. Yip, and M. A. Epton. Analysis of descriptor systems using numerical algorithms. *IEEE Transactions on Automatic Control*, AC-26(1):139–147, February 1981.

[79] G. A. Sod. A numerical study of a converging cylindrical shock. *Journal of Fluid Mechanics*, 83(4):785–794, 1977.

[80] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, 1993.

[81] J. C. Strikwerda. *Finite Difference Schemes for Partial Differential Equations*. Wadsworth and Brooks / Cole Advanced Books and Software, Pacific Grove, CA, USA, 1989.

[82] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[83] T. L. Tolliver and G. R. Zabrecky. Dynamic simulation for plant steam header control. In *Advances in Instrumentation and Control*, volume 48, pages 1323–1336, Chicago, USA, 1993.

[84] J. E. Tolsma and P. I. Barton. Efficient calculation of sparse Jacobians. *SIAM Journal on Scientific Computing*, 20(6):2282–2296, 1999.

[85] J. Unger, A. Kröner, and W. Marquardt. Structural analysis of differential-algebraic equation systems - theory and applications. *Computers and Chemical Engineering*, 19(8):867–882, 1995.

[86] R. F. van der Wijngaart. Concepts of TRIFIT, a flexible software environment for problems in two space dimensions. *Mathematics and Computers in Simulation*, 37:505–525, 1994.

[87] Y. Wada and M.-S. Liou. An accurate and robust flux splitting scheme for shock and contact discontinuities. *SIAM Journal on Scientific Computing*, 18(3):633–657, May 1997.

[88] K. A. Wilde. Boundary value solutions of the one-dimensional laminar flame propagation equations. *Combustion and Flame*, 18, 1972.

[89] E. L. Yip and R. F. Sincovec. Solvability, controllability, and observability of continuous descriptor systems. *IEEE Transactions on Automatic Control*, AC-26(3):702–707, June 1981.

Index