REDUCTION AND APPROXIMATION IN LARGE AND INFINITE POTENTIAL-DRIVEN FLOW NETWORKS

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Jason Mayes

Mihir Sen, Director

Graduate Program in Aerospace and Mechanical Engineering
Notre Dame, Indiana
April 2012
Complex systems often result in intractable mathematical models when classical methodological reduction methods are used in modeling. As a result, reduction methods more holistic in nature are normally used to avoid modeling on a component scale. In this dissertation, several new reduction methods are proposed with the intent of extending the application of classical methodological reduction methods to complex systems. As a motivating example, a large scale self-similar potential-driven tree network is used as a model complex system.

In the linear case, the self-similarity present in the physical system is translated to a self-similarity in the mathematical model. This is in turn used to analytically reduce an otherwise intractable DAE system to a much simpler ODE. It is also shown that for very large systems, it can sometimes be advantageous to approximate the system as infinite in scale. In the non-linear case, two numerical algorithms are presented to simplify dynamic analysis of piping networks. These methods are based on Chorin’s multi-step projection method for solving the Navier-Stokes equations. In addition to the self-similar tree network, other self-similar network structures are considered. In particular, grid-like networks are considered, and potential reduction methods are proposed.
Finally, in the course of studying the self-similar potential-driven tree network the appearance of fractional-order derivatives is noted several times. Based on this observation, fractional-order system identification is proposed as an extension of typical black-box reduction methods and experimental data acquired from a shell-and-tube heat exchanger is used to demonstrate its usefulness.

While the analysis presented is in terms of potential-driven transport networks, it should be noted that the specific examples used were chosen for their simplicity. But the methods used should be more generally applied to the reduction of self-similar complex systems, and even more generally, to large equation sets and DAE systems. Furthermore, the repeated appearance of fractional-order operators, and as a special case, fractional-order derivatives and integrals, suggest a very rich relationship between self-similar complex systems and fractional calculus.
For Mom and Dad.

Thanks for your love and support.

And for teaching me to finish what I start.
A.5 Integral operators ............................................. 175
A.6 Numerical implementation summary ......................... 183
A.7 The binomial theorem ......................................... 187
A.8 Binomial coefficients ......................................... 187
A.9 Other forms of the fractional-order integral and derivative . . . 189

BIBLIOGRAPHY ......................................................... 197
# FIGURES

1.1 The philosophical spectrum of analysis. ........................................ 6

2.1 Two degree of freedom translational mechanical system. ............ 18

2.2 Infinite degree of freedom translational mechanical system. ........ 19

2.3 Heat conduction is a semi-infinite body with known temperature \( \theta_w(t) \) at the boundary. The heat flux \( q''(t, 0) \) is needed. ........... 22

2.4 Finite volume approach to conduction in a semi-infinite body. .... 23

3.1 A tree network composed of operators \( L_{i,j} \) ........................... 33

3.2 An \( N = 2 \) generation tree network composed of operators \( L_{i,j} \) ... 36

3.3 A generationally independent symmetric tree network composed of operator \( L_a \) ................................................................. 43

3.4 A generationally independent asymmetric tree network composed of operators \( L_1 \) and \( L_2 \) ..................................................... 46

3.5 \( L_\infty \) shown with \( L_N \) for different values of \( N \). As \( N \) increases, \( L_N \) converges to \( L_\infty \) in the Laplace domain while in the time domain
\( L_N \) converges to \( L_\infty \). ................................................. 47

3.6 Step-response of a generation-dependent fractance device. \( \triangleright, \Diamond, \circ, \Box, \) and \( \odot \), representing \( N = 1, 2, 4, 6, \) and \( \infty \), respectively; \( E \), the \( L_2 \) error between the finite and infinite response for generation-dependent
fractance networks of varying size. ............................................. 49

3.7 Step-response of a fractance device. \( \triangleright, \Diamond, \circ, \Box, \) and \( \odot \), representing \( N = 2, 4, 6, 8, \) and \( \infty \), respectively. ......................... 50

3.8 Spring, dash-pot and Voigt elements used to model elastic and vis-
cous behavior ................................................................. 52

3.9 Infinite tree composed of springs and dash-pots used to model a
viscoelastic material. ............................................................. 54

3.10 A generationally dependent symmetric tree network with operators
\( L_i \) ........................................................................ 55
3.11 Step-response of bifurcating tree pipe networks. ◦, □, ◯, △, and solid line representing \( N = 2, 4, 8, 12, \) and \( \infty \), respectively; \( E \), the \( L_2 \) error between the finite and infinite response for fractance networks of varying size. 

3.12 A generationally dependent asymmetric tree network. 

3.13 Fractal-like network of conducting rods for heat removal from a point source. 

4.1 Schematic of tree-shaped piping network. 

4.2 Convergence test for \( P1 \) (diamonds) and \( P2 \) (circles) shows first- and second-order accuracy, respectively. 

4.3 Time-dependent flow rate due to step change in inlet pressure for networks of varying size. 

4.4 Normalized time-dependent flow rate due to step change in inlet pressure for networks of varying size. 

4.5 Time-dependent flow rate due to step change in inlet pressure for networks of varying \( \gamma \). 

4.6 Time-dependent flow rate due to positive (solid line) and negative (dotted line) step changes in inlet pressure. 

4.7 Normalized time-dependent flow rate due to positive (solid line) and negative (dotted line) step changes in inlet pressure. 

4.8 Non-dimensional inlet flow rate with \( \Delta P = 5.0 \times 10^5 \). Flow oscillates between laminar and turbulent regimes. 

4.9 Non-dimensional flow rates for each generation with \( N = 5, \gamma = 1.5, \Delta P = 5.0 \times 10^5 \). Vertical dotted lines signify instant at which transition occurs. △, ★, ◯, * and • denote flows in generations 1, 2, 3, 4 and 5, respectively. 

5.1 Schematic of self-similar network. 

5.2 Schematic of a first-generation network. 

5.3 Schematic of a second-generation network. 

5.4 Total network length vs. generation number. 

5.5 Network volume vs. generation number. 

5.6 The superposition principle used to combine the effects of four separate simple networks to produce the solution of a more complicated one. 

5.7 Flow rate vs. generation number.
5.8 Network resistance vs. generation number. 108
5.9 Critical values of beta 109
5.10 Percentage of total flow exiting networks through each sink 110
5.11 Percentage of total flow exiting networks through four sinks nearest to inlets 111
5.12 Standard deviation of the local outlet flow rates 112
5.13 Mean value of local flow rates as a function of $n$ 113
5.14 The lines of similarity in the network. The shaded area represents the portion of the network required for a solution. 115
5.15 Numbering of branches and nodes. 116
5.16 A first generation network and the $1/8$th model used for reduction. 116
5.17 A second generation network and the $1/8$th model used for reduction. 119
5.18 A third generation network. 122
5.19 The $1/8$th model of a third generation network used for analysis. 123

6.1 Step response of a black-box system. 130
6.2 Step response of Equation (6.18) and the best fit approximations for each model. Solid line with $\diamond$ is the numerical solution, dashed line marked with $\triangle$ is first-order model, dash-dot line with $\Box$ is second-order, and dotted line marked with $+$ is fractional-order model. 133
6.3 Step response of an $N$-generational fractance device composed of resistors and capacitors to a step-input in current $i(t)$. Dotted line for $N = 1$ generations, dash-dot line for $N = 3$, dashed line for $N = 6$, and solid line for $N = 9$. 134
6.4 Curve fits of numerically simulated $N$-generational fractance device. Dotted line marked with $\diamond$ denotes numerically simulated response, dashed line with $\triangle$ is for First Order model, dash-dotted line with $\Box$ for Second Order model, and solid line with $+$ for Fractional Order model. 135
6.5 Step response of Equation 6.23 and the best solutions using the three proposed models. Dotted line with $\diamond$ is the numerical solution, dashed line marked with $\triangle$ is first-order model, dash-dot line with $\Box$ is second-order, and solid line marked with $+$ is fractional-order model. 138
6.6 A simple shell-and-tube heat exchanger. 140
6.7 Raw and filtered outlet temperature step response for a shell-and-tube heat exchanger. Raw data is filtered using a 5th order Savitsky-Golay filter to remove high frequency noise. ........................................... 143

6.8 Best fit approximations to shell-and-tube data using first, second, and fractional-order models. Dotted line is filtered data, dashed line is first-order model, dash-dot is second-order model, and solid line is fractional-order model. ...................... 144

6.9 Best fit approximations to shell-and-tube data using first-, second-, and fractional-order ($\alpha = 1.8196$) models. Dotted line is filtered data, dashed line is 1st order model, dash-dot is 2nd order model, and solid line is fractional-order model with $\alpha = 1.8196$. ............. 145

7.1 Cryogen spray treatment of affected tissue. ................................. 154

A.1 The function $f(t) = t^3 - t^2$, the function $f(t + \alpha h)$, and $E^\alpha_h F_i$. Shown for both $\alpha = 0.5$ and $\alpha = 4/3$. ................................. 185

A.2 The function $f(t) = t^{2.1}$, its forward difference of order $\alpha$, and the forward difference of order $\alpha$ calculated numerically. Shown for both $\alpha = 1/3$ and $\alpha = -1/2$. ......................... 192

A.3 The function $f(t) = t^{2.1}$, its backward difference of order $\alpha$, and the backward difference of order $\alpha$ calculated numerically. Shown for both $\alpha = 0.5$ and $\alpha = -2/3$. ................................. 193

A.4 The function $f(t) = t^{2.1}$, its analytical derivative of order $\alpha$, and the derivative of order $\alpha$ calculated numerically. Shown for both $\alpha = 0.5$ and $\alpha = 4/3$. ................................. 194

A.5 The function $f(t) = t^{2.1}$, its analytical integral of order $\alpha$, and the integral of order $\alpha$ calculated numerically. Shown for both $\alpha = 0.5$ and $\alpha = 4/3$. ................................. 195

A.6 The function $f(x) = x^2$ and its fractional-order derivatives ($D^\alpha_{RL}$). Shown in (a), $\alpha = 0.2, 0.4, 0.6, 0.8, 1.0$, and in (b), $\alpha = 1, 1.2, 1.4, 1.6, 1.8, 2.0$. 196
Tables

4.1 Possible solutions for $\gamma = 1.5$
L = Laminar, O = Oscillating, T = Turbulent

4.2 Possible solutions for $\gamma = 2$
L = Laminar, O = Oscillating, T = Turbulent

4.3 Possible solutions for $\gamma = 2.5$
L = Laminar, O = Oscillating, T = Turbulent

5.1 The number of inlets, outlets, branches, and nodes for networks of increasing size

5.2 The DAE system describing the 1/8th model of a first generation network

5.3 The DAE system describing the 1/8th model of a second generation network

5.4 The DAE system describing the 1/8th model of a third generation network

5.5 Coefficients for outlet flows and total network output flow for $N = 1$ to $N = 5$ generation networks; only the first 10 outlets included for $N = 4$ and $N = 5$

6.1 Best fit constants for the three models proposed in equations (6.1-6.3) applied to the linear toy problem of equation (6.18); best fits are shown in figure 6.2; $\chi^2$ is the least squares error of the fit

6.2 Best fit constants for the three models proposed in equations (6.1-6.2) for $N = 1$, 3, 6, and 9 generation fractance devices; $\chi^2$ is the least squares error of the fit

6.3 Best fit constants for the three models proposed in equations (6.1-6.3) applied to the non-linear toy problem of equation (6.1-6.3); best fits are shown in figure 6.5; $\chi^2$ is the least squares error of the fit
6.4 BEST FIT CONSTANTS FOR THE THREE MODELS PROPOSED IN EQUATIONS (6.1-6.2) FOR THE TWO SHELL-AND-TUBE HEAT EXCHANGER DATA SETS. χ² IS THE LEAST SQUARES ERROR OF THE FIT WHEN COMPARED TO THE FILTERED DATA.
CHAPTER 1

INTRODUCTION

Reduction is the act of reducing complexity. Whether it be by assumptions, transformations, or approximations, it is fundamental in both engineering and mathematics in order to allow a complicated problem to be solved. For complex systems, which are difficult to model due to size and/or complexity, reduction is a necessity. The methods by which a complex system is reduced to a simple mathematical model can vary across a wide gamut of principles which encompasses everything ranging from methodological reductionism, where a system is broken down into its most fundamental elements, to holism, where only the overall input-output behavior of a system is of interest. Regardless of the principles, improved methods of reduction are essential for a better understanding, and ultimately, the better control of complex systems.

1.1 Reduction and simplification in engineering

In science and engineering, reduction commonly occurs in two stages: first in reducing a physical system to a mathematical model and secondly in reducing that mathematical model such that a solution can be more easily obtained. It is this second stage of reduction on which this dissertation will focus and as we speak of reduction, it will be in this context: reduction is the process of simplifying a mathematical system such that a solution is more easily obtained.
Reduction, in this sense, is already a common tool in every branch of engineering, as all analysis of physical systems is subject to forms of reduction. In most cases, analysis begins with reduction. For example, in statics or dynamics, the first step in traditional analysis is to draw a free body diagram showing all contact and non-contact forces acting on a body. Already several reductions have been made such as assuming all mass to be concentrated at a single point and various forces to be either distributed in a known fashion or being applied at a single point.

In fluids, simplifications are commonly made such as assumptions of steady state or laminar flow, or for boundary layer analysis, order of magnitude analysis ignores certain terms of Navier-Stokes to result in the boundary layer equations. Furthermore, even the full Navier-Stokes equations depend on the assumption that the fluid is a continuum, and not composed of discrete particles. And when solving the Navier-Stokes equations numerically, the Partial differential equations (PDEs) are further simplified into a set of coupled ordinary differential equations (ODEs).

Other analysis begins and ends with reduction. In heat transfer, control volume formulations result in equations dependent on parameters such as conduction and convection heat transfer coefficients. These coefficients are often derived not from first principles, but rather from experimental results and observation. The same procedure is used to create correlations for everything from Nusselt numbers to heat exchangers.

System identification can be the extreme in reduction. Oftentimes the internal components and workings of a system are completely ignored, and only the time or frequency characteristics of the system are used to construct a system model. Regarding the control of complex systems, it is often the case that a model of the system is not even necessary. But rather the entire process is reduced to a series of proportional-integral-derivative (PID) gains which can be manually tuned.
to produce a desired response to a stimulus signal. Regardless of the method, reduction (i.e., the process of reducing a physical system to a simple mathematical model) is a common occurrence in engineering analysis.

1.2 Complex systems

Trying to concretely define complexity can be a trying and complex task in and of itself. Perhaps Supreme Court Justice Potter Stewart’s famous words “I know it when I see it” offer the clearest description, as complexity means different things to people in different disciplines. Examples of complex systems could include bee swarms, financial markets, the Internet, distribution networks, biological systems, reaction-diffusion systems, and sociological systems. While all of these systems are inherently different in both nature and behavior, what they all share in common is a difficulty in modeling and it is this property that unites them.

Difficulties in modeling complex systems are the result of several factors. Size (with respect to the number of internal components or subsystems) is often an issue. As the size of a system increases, modeling becomes more difficult. Complicated couplings between all of the systems components can be difficult to capture, and as the number of components increases, a perfect model can become intractable. For example, the neurons in the brain are coupled via synapses in a fashion that can be described or modeled if the total number of neurons and connecting synapses are relatively small. In fact, the over 5600 synapses connecting the 302 neurons of C. elegans have been mapped [64], forming a very complex network of interacting neurons. However, it is estimated that the human brain consists of $10^{11}$ neurons, each with $10^3 - 10^4$ synapses coupling it with other neurons [35], making the problem of modeling the human brain completely intractable.

Other features of complex systems which create modeling difficulties include
non-linearities, system memory, and the appearance of emergent phenomena. Non-linearities can cause even the smallest perturbations to result in drastic changes in system behavior and can therefore be very difficult to accurately model. Memory is obviously a trait which increases complexity, and emergent phenomena, or behavior of a system that cannot be explained merely by a thorough understanding of its components, is a property belonging solely to complex systems. The concept of emergence first dates to the days of Aristotle, who wrote in *Metaphysics* “.. the totality is not, as it were, a mere heap, but the whole is something besides the parts ...” A similar notion was expressed by John Stuart Mill in his 1843 essay, *A System of Logic*, “The chemical combination of two substances produces, as is well known, a third substance with properties different from those of either of the two substances separately, or of both of them taken together.” In general, such phenomena in engineering systems cannot be explained by breaking a system into its component parts, but rather, emergence is instead due to the complex interactions between components. It is because these interactions are not fully understood that modeling and predicting such behavior is difficult, and as a system grows in size, the number of interactions between each of the base components grows combinatorially allowing for even more unexpected behaviors to appear.

While there are many features and properties that complex systems can exhibit, all result in a difficulty in modeling and understanding system behavior. In a special 1999 issue of *Science* devoted entirely to the idea of complex systems, researchers from a variety of fields discussed the applications of complex systems in their respective fields and offered their ideas as to what constitutes a complex system. In the call for submissions, complex systems were described as systems “whose properties are not fully explained by an understanding of its component parts [24].” Among the descriptions proposed were systems “that by design or
function [are] both difficult to understand and verify [63],” “highly structured” systems which show great “structure with variations [28],” and systems “in which there are multiple interactions between many different components [51].”

For the purpose of this work, we will adopt a somewhat limited view of complex systems and propose the following: a complex system is any system composed of a large number of components and interactions that creates intractabilities in analysis. A distinction must be made between a complex system and our mathematical representation of it. Following this view, a complex system can be described mathematically as a large system of coupled equations which are either too complex or too large to admit a sufficiently useful model or solution.

1.3 Classical reduction of complex systems

As was previously discussed, there are a variety of ways in which the process of reduction can be approached and mathematical descriptions formed. On one end of the spectrum, a system can be completely reduced to its the smallest scale and modeled using a bottom-up approach. On the other end, the inner workings and components are neglected, and only the system’s overall behavior is of interest. In the middle of these two approaches are all methods that use a combination of the two extremes. Figure 1.1 shows the range of reduction philosophies. An easy way to describe the possibilities is to think in terms of white-, grey-, and black-box approaches.

1.3.1 Methodological reductionism and the white-box

White-box reduction methods are the direct result of the philosophy of methodological reductionism. The roots of reductionism can be traced back to Descartes’ 1637 Discourse on Method, in which Descartes asserted that the world, and every-
thing in it, is like a machine. Furthermore, Descartes believed every machine was constructed of smaller mechanisms and by taking the machine apart and understanding each of its smallest mechanisms, the machine could then be fully understood when assembled again. From Descartes, the philosophy of reductionism, or the idea that complex things can be reduced to many smaller, simpler things, has been divided into several forms in both science and philosophy.

From the perspective of a scientist or engineer, methodological reductionism is its most useful form. In its clearest form, it states that the best way to understand a complex system is to first gain a clear understanding of its smallest subsystems or components and their interactions. From Descartes’ day until now, this has been the prevailing approach in science for one simple reason, “it works. It has been the key to gaining useful information since the dawn of Western science and is deeply embedded in our culture as scientists and beyond [24].”
The idea of a white-box system, in which every detail of its inner workings and construction is known, embodies this philosophy. It is a system in which a model can be constructed using a bottom-up approach, based on first principles at the smallest scale. In dynamics or controls, it is akin to writing down the equations of motion of each component to provide a model for the dynamical system. For most common textbook-type problems in engineering, it is the most useful approach. Although it must be noted that even in the simplest of textbook examples, a great deal of reduction, vis-á-vis assumptions and simplification, has occurred to even write down the equations of motion.

In a sense, an absolute reductionistic approach to analysis is impossible. No matter how small of constituent parts a system is divided into, there is always a smaller unit. For example, if a mechanical system of springs and dampers is being analyzed, writing equations of state for each mass would be one way to form a mathematical model. However, it would also be possible to write equations of motion for each molecule, atom, or sub-atomic particle of the system. Much like Xeno’s dichotomy paradox, no matter how small of constituent parts into which you divide a system, it is always possible to divide it into smaller parts, and at some point, reductionism will break down as the mathematical model creates an intractable problem.

While the reductionist approach has been shown to be successful over years of scientific progress, there are two main difficulties in its application to complex systems: size or internal complexity, and the appearance of emergent phenomena. For a complex system with a very large number of interconnected components, a reductionist approach can result in a system of equations that while solvable in principle, may be too computationally expensive to be practical. Or worse yet, the system, like the neuron model of the human brain, can be so large that reducing
it to its smallest scale becomes an impossibility. In both cases, size can lead to intractable problems.

Emergent phenomena, on the other hand, creates an entirely different sort of problem for the reductionist as the appearance of emergent phenomena is in principle contrary to the philosophy of reductionism. The reductionist’s answer to emergent phenomena must be that either the system’s behavior at the smallest scale is not correctly understood, or that there is an underlying influence at a scale even smaller than is currently understood to exist that produces the unexplainable behavior. Either way, the reductionist approach can break down when applied to the study of complex systems.

While reductionism and white-box models have been championed by intellect- tion giants such as Einstein and Newton, in recent years there has been a revolt, of sorts, against this line of thinking. Prigogine would charge that this thought process has run its course, that a reductionist approach and a dependence on deterministic equations is no longer a valid approach to science. “The more we know about our universe, the more difficult it becomes to believe in determinism. [48]”

The problem, according to Prigogine, is irreversibility, instability, and time, which can not be explained away using a reductionist approach. Instead, science must turn to statistics and probability.

The famous philosopher and economist Friedrich Hayek expressed a similar opinion in his essay “The theory of complex phenomena [7]” - that modeling of truly complex systems, such as economic and biological systems, could not be approached in same manner as simple physics. Instead, the modeling of complex phenomena using a reductionist approach cannot predict exact behavior, but can instead only be useful for predicting patterns. Despite the difference in arguments, the results are the same - the reductionist approach to modeling fails when dealing
with truly complex systems.

1.3.2 Holism and the black-box

On the other end of the spectrum, the black-box approach tends to identify more closely with the philosophy of holism than reductionism. In a holistic view of a complex system, behavior cannot be fully understood solely on the basis of an a complete understanding of its constituent parts. Instead, the behavior must be studied on the level of the system as a whole for any meaning to be deduced. While, in a certain sense, holism may seem to be the more modern philosophical approach to complex systems, it has a history far out dating that of reductionism and was first succinctly described in Aristotle's *Metaphysics* with the phrase “the whole is more than the sum of its parts.”

For a complex system where size, complexity and possibly emergent behavior prevent a reductionist approach from explaining system behavior, a holistic (or black-box) approach to analysis can allow a system to be studied and modeled as it is only concerned with input/output behavior, and not with the underlying causes of that behavior. In reality, there is very little in engineering analysis that is truly holistic. Neural networks, PID control, fuzzy logic, knowledge based systems, and other so-called expert systems which depend on human experience are excellent examples of nearly-holistic approaches. In most examples of these systems, one ofetn does not need to model the system as such, but rather chooses to do so because the details are not important. In this case, one could say “The whole is simpler than the sum of its parts” (J.W. Gibbs).

But for the large part, true holistic and emergentistic analysis is restricted to the biological fields where true emergent behaviors, such as conciousness and life, are exhibited. In these cases, the reduction from a complex system to a simple,
usable mathematical model may not be possible and the black-box approach may be the only possible method of reduction. While emergence or over-all system complexity can all be neatly swept under the rug using such an approach, it is important to remember that so called 'Black Swan' events - highly improbable and unpredictable events with significant impact [55] - can never be properly examined.

While the holistic and reductionist approaches to analysis may seem diametrically opposed, the truth is most engineering analysis contains elements of both approaches and falls somewhere between the two. Furthermore, the approach to the analysis of a complex system changes even with time. Only two hundred years ago, most chemical reactions were considered emergent phenomena as they could not be explained using the available knowledge of the day. As our understanding of the natural world increased, our ability to understand and to then mathematically describe chemical reactions developed to the point where a reductionist approach could predict the results of reactions. In general, this is the trend in science: as our knowledge base continues to expand, analysis tends to drift from the holistic to the reductionist approach.

1.3.3 Model-based system identification

Unlike white-box analysis, where all the details are known and a model can be formed from first principles, and a black-box approach, where no prior model is necessary or available, a grey-box approach allows both some model information to be deduced from a reductionist approach and some properties to be determined via holistic analysis. In systems where uncertainty in parameters and constants exists, which is the case for nearly all real physical systems, it can be possible to form a model using a reductionist approach that does a poor job of predicting system behavior even though the form of the model is correct. Rather than abandoning
the model, using a grey-box approach allows modeling uncertainty to be accounted for by using experimental data (input-output behavior) to determine constants and parameters unique to the system being modeled and tested.

For real physical systems, this is by far the most commonly used process to reduce a physical system to a mathematical model. In heat transfer, it is the procedure used to determine convection and conduction coefficients as well as Nusselt number correlations. For the control of real systems, grey-box methods are often used to determine system specific models for the purpose of model-based, model-predictive, and optimal control schemes. In a way, most control methods rely on a grey-box model. Even using PID control without a system model, methods such as Ziegler-Nichols presuppose a certain form of the plant and depend on experimental results for parameter tuning. Designing controllers using root-locus or Nyquist diagrams also requires that a system model be determined.

1.4 Extending classical reduction methods

Classical reduction methods such as white- and grey-box modeling often face difficulties when applied to complex systems. White-box modeling can result in mathematical models either too large or too complex to be useful and grey-box modeling of complex systems, which requires a parameter dependent system model, often does a poor job of describing complex behavior due to the poor choice of system model. This dissertation seeks to suggest new ways in which these classical approaches can be used and extended to allow for better mathematical models of complex systems.

In the following chapters, new methods for simplifying complex systems are used in the context of both analytical and numerical analysis. In Chapter 2, a short case will be made for using self-similarity as a reduction tool in the modeling
of complex engineering systems. It will show examples of how self-similarity can be used in reducing the mathematical model resulting from a white-box approach to system modeling. In Chapter 3, a specific example of a complex system is considered - linear self-similar potential driven tree networks - and self-similarity is used to analytically reduce the complex DAE model to a single ODE. In Chapter 4, potential driven networks are again considered, but the linear restriction is removed and other forms of reduction are used. In this example, a particular property inherent to the DAE system is used to allow for much simpler numerical simulation of a complex system. In Chapter 5, more complex networks are considered and possible methods of simplification are suggested. In Chapter 6, the repeated appearance of fractional-order solutions in earlier chapters is addressed and the use of fractional-order system identification as a black-box of reduction is suggested. Finally, we will review some of the lessons learned and identify areas of future research in which they can be used and expanded.
CHAPTER 2

USING SELF-SIMILARITY

Even a ‘simple’ complex system can result in a model that is either too large or too complex to be useful when analyzed using a reductionist approach. However, for systems with a regular, defined structure, simplification of the resulting mathematical model may sometimes be possible. Systems that exhibit some form of self-similarity are one such example and a knowledge of the similarity can give one more ‘piece’ of information that can allow for further reduction of the mathematical model. It is important to note and remember that there are many existing classical methods that take also advantage of this information, such as the use of Lie groups in traditional group theory, which offer a much more formal approach to system reduction than what is presented in this and the following chapter. These methods methods are not being considered here but do fit into the engineer’s toolbox for system reduction and should be kept in mind.

Similarity can exist in both the physical system and the mathematical level. Some complex systems, such as the human circulatory system, have a physically self-similar structure. These self-similar systems can allow a large, complex mathematical model, which results from a reductionist approach, to be reduced to a much simpler one by taking advantage of the physical structure of the system. Other systems can result in a mathematical similarity on the model level while the system itself has no physical self-similar structure. For instance, in order to find a
numerical solution to a PDE, the PDE is often first discretized and reduced to a large, or even infinite, system of similar ODEs. This large set of ODEs can sometimes be reduced analytically to single ODE by taking advantage of the similarity between each of the ODEs. The remainder of this chapter will provide examples of similarity within large equation sets that allow for reduction. The following chapter will focus on a system exhibiting a similar mathematical self-similarity that is manifested as a result of a physical self-similarity in the system.

2.1 Self-similar systems of equations

From our definition in Section 1.2, a complex system can be described mathematically as a large system of coupled equations which are either too complex or too large to admit a sufficiently useful model or solution. The goal of reduction, then, is to reduce this complex mathematical model to something simpler and more useful.

2.1.1 Reducible mathematical models

Ignoring for a moment what the system being analyzed is, a mathematical model could be of several forms with varying levels of complexity. A model could be in the form of a linear ODE or linear system of ODEs, a linear PDE or system of linear PDEs, a non-linear ODE or PDE or system of nonlinear ODEs or PDEs, or any combination of the above. Ideally, though, a model should be as simple as possible. For example, consider a complex system composed of many coupled masses. A mathematical model may be in the form of a system of linear ODEs describing the displacement of each of the masses, but if the motion of only one mass is of interest, the system of ODEs would ideally be reduced to a single ODE as the motion of the surrounding masses are not of concern.
Similarly, for systems modeled by a PDE which might not allow a simple analytical solution, the model can be reduced in complexity by reducing the PDE to a system of ODEs which can then be solved numerically. But if the entire solution is not of interest, but rather only the solution at a single point, it would be ideal to reduce the system of ODEs to a single ODE describing the solution at the point of interest. In this following sections, such a case is presented in which it is possible to reduce a PDE, first to a large system of self-similar ODEs, then to a single ODE.

Before continuing, it should be mentioned that the process of reducing a PDE to an ODE using similarity should be immediately recognizable to fluid mechanicians. One of the most famous equations of fluid mechanics, the Blasius boundary layer equation (a non-linear ODE), is derived as a similarity solution of the full Navier-Stokes equations (a system of PDEs). In this case, analysis results not just in the solution at a particular point of interest, but in fact the solution is retained at all points in the domain.

2.1.2 Reduction of infinite-order ODEs

One class of problems where reduction is possible is infinite-order ordinary differential equations. There exist certain infinite-order ODEs which can be compacted to finite-order ODEs. For example, consider the finite-order ordinary differential equation

\[ (1 + a \frac{d}{dt} + a^2 \frac{d^2}{dt^2} + a^3 \frac{d^3}{dt^3} + \cdots + a^n \frac{d^n}{dt^n}) y = f(t), \]  

(2.1)

which can be written as

\[ L[y] = f(t), \]  

(2.2)
where $L$ is the high-order operator operating on $y(t)$ in Equation (2.1). Taking the Laplace transform of Equation (2.1) gives

$$(1 + as + a^2s^2 + a^3s^3 + \cdots + a^n s^n)Y = F(s),$$

(2.3)

or

$$L[y] = F(s),$$

(2.4)

where $L$ is the Laplace transform of the operator $L$. In the Laplace domain, for Equation (2.1), $L$ is a finite geometric series which can be summed. In this case,

$$L = \sum_{k=0}^{n} (as)^k,$$

(2.5)

$$= \frac{1 - (as)^{n+1}}{1 - as}. \quad \text{(2.6)}$$

The purpose of reduction is reduce complexity, and in this case to simplify a high-order ODE to a simpler ODE. By summing the series in Equation (2.3), we can now do this. Rewriting Equation (2.3),

$$(1 - a^{n+1}s^{n+1})Y = (1 - as)F(s),$$

(2.7)

and transforming back into the time domain gives

$$y(t) - \frac{d^{n+1}}{dt^{n+1}}y(t) = f(t) - a\frac{d}{dt}f(t).$$

(2.8)

In this example, the high-order operator formed a finite geometric series in the Laplace domain. Other high-order equations could form different series or expressions that could be summed, such as harmonic, alternating, or even power series. Furthermore, infinite-order operators can often be reduced in the same manner. For example, if the previous example, for $n = \infty$, the series in Equation (2.3) can still be summed for certain values of $as$. In this case, for $as < 1$, which corresponds
to low frequencies in the time-domain, the operator $\mathcal{O}_L$ can still be summed exactly to

$$\mathcal{O}_L = \frac{1}{1 - as}, \quad (2.9)$$

reducing the infinite-order ODE

$$(1 + a \frac{d}{dt} + a^2 \frac{d^2}{dt^2} + a^3 \frac{d^3}{dt^3} + \cdots + a^n \frac{d^n}{dt^n} + \cdots) y(t) = f(t), \quad (2.10)$$

to a much simpler ODE

$$y(t) = f(t) - a \frac{d}{dt} f(t). \quad (2.11)$$

2.1.3 Reduction of infinite sets of ODEs

The reduction of operators from infinite-order to finite-order is only useful in the practical sense if infinite-order differential equations occur in the modeling of real systems. The question then arises, what kind of system results in an infinite-order differential equation, or even a high-order equation? Any system that can be described by an infinite or very large system of coupled linear ODEs can result in an infinite or high-order equation. Any system of coupled linear ODEs can be reduced to a set of coupled first-order ODEs and any set of coupled first-order ODEs can be reduced to a single higher-order differential equation. This methodology is often used in linear controls, where this higher-order ODE is represented in the frequency domain by a transfer function, commonly denoted $G(s)$, which expresses the input/output relationship of a system.

Sets of coupled, linear ordinary differential equations can be reduced to a single ODE and it is common practice in linear controls to do so. A frequently encountered example from a linear controls textbook is shown in Figure 2.1, where a force is applied to $m_1$ and its motion, $x_1(t)$, is desired.
The equations of motion for the coupled mass problem shown in Figure 2.1 are

\[
m_1 \ddot{x}_1 = f(t) - k_1 (x_1 - x_2), \quad (2.12)
\]
\[
m_2 \ddot{x}_2 = k_1 (x_1 - x_2) - k_2 (x_2), \quad (2.13)
\]

which can be written as set of four first-order equations,

\[
\begin{bmatrix}
    \dot{x}_1 \\
    \dot{x}_2 \\
    \dot{x}_2
\end{bmatrix}
= \begin{bmatrix}
    0 & 1 & 0 & 0 \\
    -\frac{k_1}{m_1} & 0 & \frac{k_1}{m_1} & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \dot{x}_1
\end{bmatrix}
+ \begin{bmatrix}
    \frac{1}{m_1} \\
    0 \\
    0
\end{bmatrix}
\begin{bmatrix}
    f(t)
\end{bmatrix}.
\]

(2.14)

From the first-order system of equations in Equation (2.14), it is a straightforward task to compute the transfer function between \( x_1(t) \) and \( f(t) \),

\[
G(s) = \frac{X_1(s)}{F(s)} = \frac{m_2 s^2 + k_1 + k_2}{m_1 m_2 s^4 + (m_1 k_1 + m_1 k_2 + m_2 k_1) s^2 + k_1 k_2},
\]

(2.15)

(2.16)
which is just the Laplace transform of the fourth-order differential equation

\[
(m_1m_2 \frac{d^4}{dt^4} + (m_1k_1 + m_1k_2 + m_2k_1) \frac{d^2}{dt^2} + k_1k_2) x_1(t) = (m_2 \frac{d^2}{dt^2} + k_1 + k_2) f(t).
\]

(2.17)

It should quickly be noted that this equation could also be derived without the use of the Laplace transform by simply differentiating Equation (2.12) twice and substituting (2.13), but the reason for using the Laplace transform method will become more obvious in the following chapter.

If more masses are added in series to the mechanical system in Figure 2.1, the system of equations grows larger, but the same procedure will still yield a single scalar differential equation of even higher-order. An infinite system of springs and masses would then result in an infinite-order differential equation.

For example, for the infinite system shown in Figure 2.2, the resulting system of first-order equations is infinite dimensional. If each of the masses, \(m_1, m_2, \ldots m_\infty\) are all taken to be \(m\) and the spring constants are all assumed \(k\), then the infinite system of equations can be reduced, in the Laplace domain, to give the transfer function

\[
\frac{F(s)}{X_1(s)} = \left(\frac{k^2}{ms^2 + k - k^2} - \frac{k^2}{ms^2 + 2k - \cdots} \right).
\]

(2.18)
which could alternatively be written as a ratio of two polynomials in the Laplace variable $s$, each of infinite-order, corresponding to an infinite-order differential equation between $x_1(t)$ and $f(t)$. However, using the continued fraction form of the transfer function in Equation (2.18), the infinite-order differential equation can be reduced to a simpler finite-order ODE that can then be solved for $x_1(t)$. In this case, the continued fraction in (2.18) can be simplified so that

$$\frac{F(s)}{X_1(s)} = \frac{ms^2}{2} + \frac{s}{2} \sqrt{m^2s^2 + 4mk}. \quad (2.19)$$

Incidentally, this example system could be reduced by another method: by taking the system of ODEs and reducing it to the PDE of linear elasticity. In this case, the usefulness of either method of reduction ultimately depends on the person performing the reduction. In some instances, the PDE which contains the full solution at every point may be more useful, while for others the ODE describing only the motion of the first mass may prove to be more advantageous. In both cases, reduction is, ultimately, the process of simplifying an otherwise complex mathematical model such that a solution is more easily obtainable.

As was shown in Section 2.1.2, the same procedure could be followed to reduce a very large set, as opposed to an infinite set, of first-order ODEs, since any finite continued fraction can be expressed as a finite sum. In cases where only one state is of interest, this can be a very useful reduction, reducing the number of ODE’s, and thus states, to be solved for to just one. And while no real system is ever truly infinite, the idea of an infinite system still has great value. For extremely large systems, in our example where the number of masses is finite but very large, the order of the resulting scalar differential equation is very high. But as the size of the system increases, the solution converges to the solution of the infinite system so assuming a very large system to be infinite can often lead to a very good approximation.
2.1.4 Reduction of PDE’s

Partial differential equations can be reduced to an infinite number of ordinary differential equations by a variety of methods including finite-difference, finite-volume, spectral, and finite element methods. Once a PDE has been reduced to a system of ODEs it may then be possible to reduce that system of ODEs to a single higher-order ODE. In certain cases, it is then possible to reduce that high or infinite-order ODE to a single finite-order ODE. Usually, these cases involve systems that exhibit some sort of self-similarity between the resulting ODEs. Using finite difference or finite volume expansions of a PDE is one such case when similarity within the resulting set of ODEs can be taken advantage of.

For example, consider the heat equation

\[ \frac{\partial \theta}{\partial t} = \alpha \frac{\partial^2 \theta}{\partial x^2}, \]  

(2.20)

with the boundary and initial conditions

\[ \theta(t, 0) = \theta_w(t), \]  

(2.21)

\[ \theta(0, x) = 0, \]  

(2.22)

\[ \lim_{x \to -\infty} \theta(t, x) < \infty, \]  

(2.23)

describing one-dimensional heat conduction in a semi-infinite body as shown in Figure 2.3, where the heat flux \( q''(t, 0) \) is desired. This problem has previously been examined by Podlubny [47] as motivation for the use of fractional calculus, although his solution methodology is very different. In Podlubny’s example, the heat flux from a blast furnace is desired so the operation of the furnace can be monitored. The furnace is modeled as a semi-infinite body and the semi-heat equation is derived relating the heat flux at the boundary with the order 1/2 derivative of the wall temperature.
As was the case in Section 2.1.3, the entire solution, in this case $\theta(x,t)$, is not necessary. Instead, we only need the flux at the furnace wall. By using a finite volume approach and dividing the domain into an infinite number of volumes as shown in Figure 2.4, the PDE (2.20) can be reduced to the infinite set of ODEs

$$\Delta x \frac{d\theta_i}{dt} = q''(t,0) - \frac{\alpha}{\delta x} (\theta_i - \theta_{i-1}), \quad (2.24)$$

for $i = 0$, and

$$\Delta x \frac{d\theta_i}{dt} = \frac{\alpha}{\delta x} (\theta_{i+1} - \theta_i) - \frac{\alpha}{\delta x} (\theta_i - \theta_{i-1}), \quad (2.25)$$

for $i = -1, \ldots, -\infty$, with boundary and initial conditions

$$\theta_0(t) = \theta_w(t), \quad (2.26)$$

$$\theta_i(0) = 0, \quad (2.27)$$

$$\theta_{-\infty} = \theta_{-\infty+1}. \quad (2.28)$$
In the Laplace domain, the infinite system of equations is

\[ s\theta_w = \frac{Q''(s,0)}{\Delta x} - \frac{\alpha}{(\delta x)^2}(\theta_w - \theta_{-1}), \]  

(2.29)

and

\[ s\theta_i = \frac{\alpha}{(\Delta x)^2}(\theta_{i+1} - 2\theta_i + \theta_{i-1}), \]  

(2.30)

\[ \theta_{-\infty} = \theta_{-\infty + 1}, \]  

(2.31)

for \( i = -1, \ldots, -\infty + 1 \), after all boundary and initial conditions have been applied. Combining and rearranging the equations gives the self-similar equation set

\[ (s + \beta)\theta_w = \frac{Q''(s,0)}{\delta x} + \beta \theta_{-1}, \]  

(2.32)

\[ (s + 2\beta)\theta_{-1} = \beta(\theta_w + \theta_{-2}), \]  

(2.33)

\[ \vdots \]

\[ (s + 2\beta)\theta_{-\infty + 2} = \beta(\theta_{-\infty + 3} + \theta_{-\infty + 1}), \]  

(2.34)
\[(s + \beta)\theta_{-\infty+1} = \beta\theta_{-\infty+2}, \quad (2.35)\]

where \(\beta = \frac{\alpha}{(\delta x)^2}\). Reduction of this system of algebraic equations to a single equation for \(Q''(s,0)\) is now a matter of algebra. Solving Equation (2.35) for \(\theta_{-\infty+1}\),

\[\theta_{-\infty+1} = \frac{\beta}{s + \beta}\theta_{-\infty+2}, \quad (2.36)\]

and then substituting into Equation (2.34) to solve for \(\theta_{-\infty+2}\), gives

\[\theta_{-\infty+2} = \frac{\beta}{s + 2\beta - \frac{\beta^2}{s + \beta}}\theta_{-\infty+3}. \quad (2.37)\]

Repeating the process an infinite number of times yields the result

\[(s + \beta - \frac{\beta^2}{s + 2\beta - \frac{\beta^2}{s + 3\beta - ...}})\theta_w = \frac{Q''(s,0)}{\delta x}. \quad (2.38)\]

The operator \(\mathcal{O}_L\), taken to be

\[\mathcal{O}_L = (s + \beta - \frac{\beta^2}{s + 2\beta - \frac{\beta^2}{s + 3\beta - ...}}), \quad (2.39)\]

corresponds to the transfer function between the desired heat flux \(q''(t,0)\) and the wall temperature \(\theta(t,0)\). It can also be written in the form

\[\mathcal{O}_L = \frac{a_0 + a_1s + a_2s^2 + \cdots + a_3s^3}{b_0 + b_1s + b_2s^2 + \cdots + b_3s^3}, \quad (2.40)\]

using a variation of Euler’s continued fraction formula, which then results in the infinite-order ODE

\[(a_0 + a_1\frac{d}{dt} + a_2\frac{d^2}{dt^2} + \cdots + a_3\frac{d^3}{dt^3})\theta_w(t) = (b_0 + b_1\frac{d}{dt} + b_2\frac{d^2}{dt^2} + \cdots + b_3\frac{d^3}{dt^3})q''(t,0). \quad (2.41)\]

However, the continued fraction representation of Equation (2.39) allows for even further simplification. The continued fraction in \(\mathcal{O}_L\) can be reduced to

\[\mathcal{O}_L = \frac{s}{2} + \frac{1}{2}\sqrt{s^2 + 4\beta s}. \quad (2.42)\]
Now rewriting Equation (2.38) using (2.42) as

\[
Q''(s, 0) = \delta x \left( \frac{s^2}{2} + \frac{1}{2} \sqrt{s^2 + 4\beta s} \right) \theta_w(s), \tag{2.43}
\]

\[
= \left( \delta x \frac{s}{2} + \frac{1}{2} \sqrt{(\delta x)^2 s^2 + (\delta x)^2 4\alpha s} \right) \theta_w(s) \tag{2.44}
\]

and taking the limit as \( \delta x \to 0 \),

\[
Q''(s, 0) = \sqrt{\alpha} s^{1/2} \theta_w(s). \tag{2.45}
\]

The inverse Laplace transform then gives

\[
q''(t, 0) = \sqrt{\alpha_0} \frac{D_t^{1/2}}{2} \theta_w(t), \tag{2.46}
\]

in the time domain, where \( \_0D_t^{1/2} \) is the Caputo fractional derivative operator of order \( \alpha = 1/2 \). For a detailed development of the fractional derivative operator and its numerical implementation, refer to Appendix A.4.

Another alternative to Equation (2.46) is to either numerically solve Equation (2.20) for \( \theta(t, x) \) and then numerically compute

\[
q''(t, 0) = \frac{d\theta}{dt}(t, 0), \tag{2.47}
\]

or to find an analytical solution to (2.20) and differentiate in the same manner. In either case, the entire solution is necessary when only the heat flux at the wall is needed. As for the option of pursuing an analytical solution, it was previously mentioned in regards to the case of the boundary layer equations, similarity solutions do exist to reduce heat equation (a PDE) to an ODE in which case the solution can be obtained at all points in the domain. In such cases it is possible that such a reduction can be beneficial to the whomever is performing the reduction, provided the result of the reduction is that a solution is more easily obtainable with respect to the individual performing the reduction.
While any linearly independent set of \( N \) linear ODEs in \( N \) variables can be written as a single high-order scalar ODE, the ability to compact this high-order ODE to a simpler, lower order ODE depends on finding a series representation of the high dimensional operator in Laplace space. For PDEs which have been reduced to a set of ODEs using finite differences or finite volumes, this is often possible due to the similarity that exists between the resulting ODEs. While this method of reduction does not provide the entire solution to the PDE, there are often cases where only a boundary condition or the solution at a point is needed. In the example of Section 2.1.4, only the heat flux at the wall is needed, so to compute the solution at every point is only adding complexity.

Furthermore, reductions of this type could have uses when solving PDEs with difficult boundary conditions. In the previous example, Equation (2.46) provides a relation to transform the Dirichlet boundary condition

\[
\theta(t, 0) = \theta_w(t)
\]

(2.48)

to the Neumann condition

\[
\frac{d\theta}{dx}(t, 0) = \sqrt{\alpha_0} D_1^{1/2} \theta_w(t),
\]

(2.49)

and using this method, relations can be derived for other PDEs and boundary conditions.

### 2.2 Other similarity methods

One should of course realize that there are many other possible ways to use self-similarity in the reduction of a complex system. Several other methods will be discussed within the context of a particular network topology are discussed in Chapter 5. Additionally, there is an entire branch of Group Theory dedicated to doing just that. Similarity groups, and in particular, Lie Groups, can be used to
reduce physically self-similar systems, self-similar equation sets, ODEs and PDEs in a similar manner as has been done in this chapter, but using a completely different approach. It is a much more formal approach to reduction and one which is on the fringes of everything discussed within this dissertation, but is not directly used. Nevertheless, it must still be mentioned.

One of the key differences between the Lie Group approach and what is done in this and the following chapters lies in the types of similarity considered. The symmetry that Lie Groups seek to exploit is more of a geometrical nature. For example, a snow-flake can be imagined as a symmetrical $2 - D$ object with 12 discrete similarity operations. Under each of these distinct operations, the snow-flake is considered invariant - that is, the snow-flake after the operation cannot be distinguished from the snow-flake before the operation. Similarly, transformations for PDEs or ODEs can be found that can reduce the complexity of the original equation (perhaps by reducing a PDE to an ODE), but that in no way change the equation being solved - that is to say the equations being reduced are invariant to the transformation. These transformations take advantage of geometrical symmetries within the system.

An excellent example of the Group Theory approach to reduction of a complex systems is the Blasius boundary layer equations [9]. In this example two coupled PDEs are first tranformed into a single PDE by introducing the stream function. That PDE is then reduced to a third-order ODE through the use of a similarity variables determined through the use of dimensional analysis. Furthermore, the Lie group of the Blasius equation can then be used to further reduce it in a way that the third-order ODE can be reduced to a first-order ODE. The most incredible part of this process is that no information is lost as the original system is invariant to the transformations applied. Thus the solution to the PDE system at the beginning
is given, at all points, by the solution of the first-order ODE resulting from the reduction. This is much more powerful than the approach taken in this and the following chapters, but it is not always necessary and is not always possible. Thus the methods presented are not intended as a replacement in any sense, but rather as an extension and an alternative when such methods may fail.

2.3 Using physical similarity

In Section 2.1, similarity within a large set of ODEs resulting from the discretization of a PDE is used to reduce the complexity of a mathematical model. Similarly, a physical self-similarity can lead to a regular coupling between equations and can be used to reduce the complexity of a mathematical model of a physical system. In the following chapter, a physically self-similar transfer network is proposed and both its physical structure and the behavior of its component parts are used to deduce the overall behavior of the entire system.
CHAPTER 3

ANALYTICAL APPROXIMATION OF TREE NETWORKS

When modeling a self-similar system using a white-box approach, it is likely that the resulting mathematical model will exhibit some form of similarity as well. In this chapter, a self-similar system is examined that produces such a model and a method of reduction is applied to this model to aid in analysis. For the interested reader, the majority of the contents of this chapter is available in [40].

3.1 Potential driven transport networks

Potential-driven transport networks, such as mammalian circulatory and bronchial systems, Heating, Ventilating and Air Conditioning systems, city plumbing networks, and power distribution grids, frequently appear in many branches of science and engineering. While the nature of these systems varies, the modeling of each network produces a system of differential algebraic equations (DAEs) governing the flow. For small-scale systems with only a few branches and nodes, there are several techniques to solve the DAE system. But large-scale systems, or systems with a large number of nodes and branches, result in large systems of DAEs that can be expensive to solve.

There are many techniques that can be used for steady-state analysis of small-scale transport networks. The Cross method [15] of analyzing fluid piping networks
and Kirchoff’s circuit laws for analyzing electrical circuits are two such techniques. But dynamic analysis of even a small network is much more difficult and requires the solution of the DAE system. For large-scale transport networks even steady-state analysis can become complicated due to the extremely large number of equations in the resulting mathematical model. In the same manner, as the scale and complexity of a transport network increase, time-dependent solutions can be increasingly difficult to obtain and the problem can become effectively intractable due to its size.

DAE systems are often classified by their index and while different definitions exist, all are a measure of the difficulties one can expect in calculating a solution. More accurately, the term index is used to denote the ‘distance’ between a DAE system and an ODE system \([4, 8]\). For large, high-index DAE systems, such as those that arise from transport networks, there are methods to reduce the index \([29]\). But at best the result of such methods is a large system of ODEs that must still be reduced or simplified. Oftentimes this reduction is in the form of assumptions or approximations, such as assuming one-dimensional flow or using linear approximations when modeling. Other methods of reduction work to reduce the size of the mathematical model. For example, a complex circuit can be reduced to a much simpler macromodel of the original circuit using Thévenin’s and Norton’s equivalent circuits. This reduces the number of equations in a mathematical model and can greatly reduce the computational expense of obtaining time-dependent solutions.

Similarly, some mathematical models can be reduced by taking advantage of physical structure or self-similarity in the original network to create a simpler network model. Nakagawa and Sorimachi \([43]\) do this in the case of an infinite resistor-capacitor (RC) circuit with a physically repeated pattern to create a sim-
pler, finite macromodel circuit, allowing the dynamic solution of a previously in-
tractable mathematical problem. Similar approaches have been used with other
infinite electrical networks, including networks with differing forms of similarity
such as ladders, grids, and rings [34,39,53,58,65]. In each of these cases, models
are reduced by taking advantage of physical structure and the infinite nature of
the system.

All of these methods of reduction and simplification seek to reduce complexity
in such a way that solutions can be obtained. In most cases, a system is reduced by
approximating it as something smaller in scale or simpler in nature. But reduction
of a network can also be accomplished by approximating it as a network even
larger in scale if that approximation simplifies it in a way that allows solutions to
be obtained more easily. In the remainder of this chapter, large-scale systems are
reduced by approximating them by their infinite extensions which paradoxically
allows for simpler solutions. Potential-driven flows in bifurcating tree networks are
considered as examples to show how the system model can be reduced from a very
large or infinite set of DAEs to a single ODE.

The self-similar potential driven tree system seen in the arterial network will
be the complex system modeled in this chapter, although there are surely ways to
extend this to other self-similar models, such as self-similar grids and ladders as
seen in [31,39]. The tree model was chosen for several reasons. First of all, it is
surprisingly common and has applications in a variety of different fields [6,11,38].
To develop a solution that can be applied across such a broad spectrum of appli-
cations requires that the problem be approached from a very general perspective.
Secondly, the behavior of these systems can also be interesting in and of them-
selves. Systems of this nature frequently exhibit fractional-order responses, even
when constructed entirely of integer-order components, and are used in some fields
as a way to physically model fractional-order behavior.

Analyzing the potential driven tree network using a reductionist approach will result in a mathematical model in the form of a very large system of differential algebraic equations (DAEs) consisting of a differential transfer equation for each branch and a conservation equation for each node. Due to the physical self-similar structure of the network, the system of DAEs will exhibit a form of similarity which will allow for reduction in the same manner as the PDE of Section 2.1. Reduction of the mathematical model will be based on the premise that only the behavior of the network as a whole are of interest, and not the local flow rates or pressures at any internal branches or nodes.

It has been previously mentioned that self-similar potential flow trees are amazingly common to both natural and engineered systems. Something must be said about the frequency with which these systems are encountered and to do this the work of Bejan [6] must be mentioned. The function of most tree flows is to connect a single point (source or sink) to an infinity of points (volume or area) with the purpose of delivering or transferring some flow quantity. It is no coincidence that these systems are so common. Bejan has studied the optimization of many different flow systems to better understand the systems we see in nature and has found that the tree system seen in nature is not a random or meaningless design, but is the result of the optimization of point-to-volume or volume-to-point flow systems subject to volume constraints. By using a single deterministic principle, global maximization of performance subject to constraints, Bejan is able to reproduce or predict the patterns seen in natural systems. The self-similar tree then, in addition to being a commonly occurring design in nature, has applications as well in man-made flow systems as the most efficient or optimized structure for point-to-volume or volume-to-point flow delivery systems.
3.2 Trees

Bifurcating tree networks, also referred to as perfect binary trees and shown in Figure 3.1, regularly occur in both natural and artificial engineering systems. Examples include biological transport networks, river basin drainages, viscoelasticity models, and microchannel electronic cooling systems [1, 11, 13, 16, 19, 20, 23, 30, 38, 45, 56, 61]. Its appearance in so many natural and man-made systems is no coincidence; Bejan [6, 18] has shown the bifurcating tree network geometry to be the optimized result of point-to-volume flow systems.

This network begins with a bifurcation into two branches, which make up the first generation. Each branch in the first generation then splits into two new branches in the following generation. This repeats itself for each successive generation. A network of \( N \) generations then results in \( 2^{N+1} - 2 \) total branches and
2^N − 2 branching points. Each branch can be represented uniquely with a pair of integers \((i, j)\) where \(1 \leq i \leq N\) is the generation number the branch is located in, and \(1 \leq j \leq M_i = 2^i\) is the branch number within that generation.

The system studied is assumed to have three basic properties: flow (of fluid, heat, energy, etc.) occurs through the network; the transfer is driven by a potential difference across the system; and the transferred quantity is conserved at each of the branching points. With these assumptions, a transfer equation can be written for every branch of the form

\[ L_{i,j}q_{i,j}(t) = \Delta p_{i,j}(t), \]  

(3.1)

where \(q_{i,j}(t)\) is the transfer through branch \((i, j)\), \(\Delta p_{i,j}(t)\) is the potential difference across the branch \((i, j)\), and \(L_{i,j}\) is the operator relating the two. The operators \(L_{i,j}\) could each be of any form, the only restriction made is that they be linear and, if necessary, restricted by appropriate initial conditions such that a unique inverse exists. For the operator \(L\) and the equation \(Lx = y\), the solution \(x\) is given by

\[ x = L^{-1}y + \sum \alpha_n \phi_n, \]  

(3.2)

where \(L^{-1}\) is the inverse of \(L\), \(\alpha_n\) is a constant, and \(\phi_n\) belongs to the null space of the operator \(L\), i.e. \(L\phi_n = 0\). For many engineering systems, either the kernel of the operator is trivial or initial conditions can be chosen such that \(\sum \alpha_n \phi_n = 0\). In either case, the rightmost term in Equation (3.2) can be disregarded and we will continue to work under these assumptions. With these conditions, the inverse operator \(L^{-1}\) is unique.

In addition, a conservation equation can be written for each branching point within the network of the form

\[ q_{in}^n = \sum_{k=1}^{P} w_k q_{out}^k, \]  

(3.3)
where $P$ is the number of branches leaving each branching point and $w_k$ is a weighting factor.

For any bifurcating tree network this results in a set of $3 \times 2^N - 4$ equations. In its most general form, this DAE system can be written as

$$A \frac{dq}{dt} + Eq = f(t),$$

where $q$ is the vector of flow rates and $f(t)$ contains $\Delta p$, the vector of potential differences. Because some of the equations in this coupled system are algebraic, it must be noted that the matrix $A$ is guaranteed to not be full rank and is thus not invertible. This is, by definition, a differential-algebraic system. Additionally, the matrix $E$, while relatively sparse, is not banded. As the size of the network grows, the resulting system of DAEs increases exponentially in size and complexity. For large scale systems, with $N$ very large, reduction then becomes not just useful, but necessary to obtain solutions. The goal is to approximate this large or infinite set of DAEs by a single ODE

$$L_Nq(t) = \Delta p(t),$$

where $q(t)$ is the total transfer through or across the network, $\Delta p(t)$ is the potential difference across the network, and $L_N$ is the approximate operator relating the potential difference and induced transfer. This will provide a reduced mathematical model to allow for dynamic solutions of even very large systems.

3.2.1 Reduction in the time domain

In an effort to reduce the mathematical model of a transport network, the goal is then to find the operator $L_N$ in Equation (3.5) which is equivalent to the complete tree composed of the operators $L_{i,j}$ that describe the flow through each
branch. The system of equations can, however, be simplified by noting the self-
similar structure present in both the network and the resulting equations, and using
that observation to reduce the size of the equation set and eliminate unnecessary
unknowns. An equation of the form shown in Equation (3.1) exists for every branch
within the network. These equations can be added along any possible path from
network inlet to outlet to eliminate the intermediate potential differences. For
example, consider the 2-generation network shown in Figure 3.2.

\begin{align*}
L_{1,1}q_{1,1} &= \Delta p_{1,1} = p_{in} - p_{1,1}, \\
L_{1,2}q_{1,2} &= \Delta p_{1,2} = p_{in} - p_{1,2}, \\
L_{2,1}q_{2,1} &= \Delta p_{2,1} = p_{1,1} - p_{out}, \\
L_{2,2}q_{2,2} &= \Delta p_{2,2} = p_{1,1} - p_{out},
\end{align*}

Figure 3.2: An $N = 2$ generation tree network composed of operators $L_{i,j}$

There are $2^N$ (with $N = 2$) possible paths through this network. The six
transfer equations for this system (one for each branch) are

\begin{align*}
L_{1,1}q_{1,1} &= \Delta p_{1,1} = p_{in} - p_{1,1}, \\
L_{1,2}q_{1,2} &= \Delta p_{1,2} = p_{in} - p_{1,2}, \\
L_{2,1}q_{2,1} &= \Delta p_{2,1} = p_{1,1} - p_{out}, \\
L_{2,2}q_{2,2} &= \Delta p_{2,2} = p_{1,1} - p_{out},
\end{align*}
\[ L_{2,3}q_{2,3} = \Delta p_{2,3} = p_{1,2} - p_{\text{out}}, \quad (3.6e) \]
\[ L_{2,4}q_{2,4} = \Delta p_{2,4} = p_{1,2} - p_{\text{out}}. \quad (3.6f) \]

where \( q_{i,j}, \Delta p_{i,j}, \) and \( p_{i,j} \) are functions of time as before. Additionally, assuming unit weights the conservation equations for the two nodes are

\[ q_{1,1} = q_{2,1} + q_{2,2}, \quad (3.7a) \]
\[ q_{1,2} = q_{2,3} + q_{2,4}. \quad (3.7b) \]

Finally, the total flow, \( q \), through the simplified network is given by

\[ q = q_{1,1} + q_{1,2}. \quad (3.8) \]

and \( L_2 \) is the operator describing the behavior of the simplified 2-generation tree in

\[ L_2q = \Delta p. \quad (3.9) \]

By combining the transfer equations along the four unique paths from inlet to outlet, the interior potentials, \( p_{1,1} \) and \( p_{1,2} \), are eliminated to yield four new equations

\[ L_{1,1}q_{1,1} + L_{2,1}q_{2,1} = p_{\text{in}} - p_{\text{out}} = \Delta p, \quad (3.10a) \]
\[ L_{1,1}q_{1,1} + L_{2,2}q_{2,2} = p_{\text{in}} - p_{\text{out}} = \Delta p, \quad (3.10b) \]
\[ L_{1,2}q_{1,2} + L_{2,3}q_{2,3} = p_{\text{in}} - p_{\text{out}} = \Delta p, \quad (3.10c) \]
\[ L_{1,2}q_{1,2} + L_{2,4}q_{2,4} = p_{\text{in}} - p_{\text{out}} = \Delta p. \quad (3.10d) \]

Equations (3.10) can then be manipulated to solve for the \( q_{2,j} \)'s by moving the leftmost term to the right side and then operating on both sides by \( L_{2,j}^{-1} \), which is the inverse operator that when applied to \( L_{2,j} \) produces the identity operator, i.e.,
\[ L_{i,j}^{-1}L_{i,j} = I. \] The \( q_{2,j} \)’s are then

\[
q_{2,1} = L_{2,1}^{-1}(\Delta p - L_{1,1}q_{1,1}), \quad (3.11a)
\]

\[
q_{2,2} = L_{2,2}^{-1}(\Delta p - L_{1,1}q_{1,1}), \quad (3.11b)
\]

\[
q_{2,3} = L_{2,3}^{-1}(\Delta p - L_{1,2}q_{1,2}), \quad (3.11c)
\]

\[
q_{2,4} = L_{2,4}^{-1}(\Delta p - L_{1,2}q_{1,2}). \quad (3.11d)
\]

Using Equations (3.7) these equations combine to give

\[
q_{1,1} = \{L_{2,1}^{-1} + L_{2,2}^{-1}\}(\Delta p - L_{1,1}q_{1,1}), \quad (3.12a)
\]

\[
q_{1,2} = \{L_{2,3}^{-1} + L_{2,4}^{-1}\}(\Delta p - L_{1,2}q_{1,2}). \quad (3.12b)
\]

The process is then repeated to solve for and eliminate the \( q_{1,j} \)’s by first applying the inverse operators \( \{L_{2,1}^{-1} + L_{2,2}^{-1}\}^{-1} \) and \( \{L_{2,3}^{-1} + L_{2,4}^{-1}\}^{-1} \) to Equation (3.12), respectively, to give

\[
\{L_{2,1}^{-1} + L_{2,2}^{-1}\}^{-1}q_{1,1} = \Delta p - L_{1,1}q_{1,1}, \quad (3.13a)
\]

\[
\{L_{2,3}^{-1} + L_{2,4}^{-1}\}^{-1}q_{1,2} = \Delta p - L_{1,2}q_{1,2}. \quad (3.13b)
\]

Collecting the \( q_{1,j} \)’s on the left and left operating with \( \{L_{2,1}^{-1} + L_{2,2}^{-1}\}^{-1} + L_{1,1} \) and \( \{L_{2,3}^{-1} + L_{2,4}^{-1}\}^{-1} + L_{1,2} \), respectively, results in

\[
q_{1,1} = \{L_{2,1}^{-1} + L_{2,2}^{-1}\}^{-1} + L_{1,1} \Delta p, \quad (3.14a)
\]

\[
q_{1,2} = \{L_{2,3}^{-1} + L_{2,4}^{-1}\}^{-1} + L_{1,2} \Delta p, \quad (3.14b)
\]

which can then be substituted into Equation (3.8), resulting in the single equation

\[
q = \{L_{2,1}^{-1} + L_{2,2}^{-1}\}^{-1} + \{L_{2,3}^{-1} + L_{2,4}^{-1}\}^{-1} \Delta p. \quad (3.15)
\]

For the engineer or scientist familiar with circuit analysis, it might be noticed that this equation could also be derived by a simple superposition of the various components in series and parallel. But for others, it may not be so obvious.
Rewriting Equation (3.15) in the form of (3.9), the system operator for a bifurcating network with \( N = 2 \) generations can be given as

\[
L_2 = \left\{ \left( L_{1,1} + \{ L_{2,1}^{-1} + L_{2,2}^{-1} \} \right)^{-1} + \left( L_{1,2} + \{ L_{2,3}^{-1} + L_{2,4}^{-1} \} \right)^{-1} \right\}^{-1}. 
\] (3.16)

This same process can be repeated for a tree network with \( N \) generations. The \( 2^{N+1} - 2 \) branch equations can be summed along all of the \( 2^N \) possible unique paths through the network to eliminate all of the intermediate potentials. The conservation equations can then be used to eliminate the unknown flow rates within the network, beginning with the \( N^{th} \) generation and progressively working back to the first, just as was done before. For a large \( N \), this process results in the reduction of a very large set of equations to a single equation in terms of the overall flow rate and potential difference. For an \( N \)-generational potential driven tree system, \( L_N \) can be written as

\[
L_N = \left\{ \sum_{j=1}^{2} \left( \sum_{k=1}^{2} \{ L_{2,2(j-1) + k} \} + \sum_{l=1}^{2} \{ L_{3,2(k-1) + l + \ldots} \}^{-1} \right)^{-1} \right\}^{-1}. 
\] (3.17)

3.2.2 Reduction in the frequency domain

The usefulness of Equation (3.17) depends on how easily \( L_N \) can be evaluated. For algebraic operators, computation of the system operator is simple as the inverse operators are merely the algebraic inverse of the operator. But in the case of integral or differential operators, evaluation of (3.17) can be complicated even when it is assumed that all the inverse operators are unique. As many dynamic analysis of engineering systems is concerned with differential operations, this severely limits the utility of (3.17). However, while evaluation of \( L_N \) in the case of differential
or integral operators may be difficult in the time domain, it can be calculated relatively easily in the frequency domain by use of the Laplace transform.

Consider the same two-generation system as shown before in Figure 3.2. If it is assumed that each of the operators $L_{i,j}$ are differential, then the set of differential algebraic equations (3.6)–(3.9) results. Taking the Laplace transform of this set of DAEs and following the same procedure as before, the system operator can be written as

$$L_N = \left\{ \sum_{j=1}^{2} \left\{ \sum_{k=1}^{2} \left\{ \sum_{l=1}^{2} \left\{ L_{3,2(k-1)+l}^{-1} \right\}^{-1} \right\}^{-1} \right\}^{-1} \right\}^{-1}.$$  

(3.18)

where the notation $L_{i,j} = \mathcal{L}[L_{i,j}]$ denotes the Laplace transform of the operator $L_{i,j}$.

Furthermore, for differential or integral operators $L_{i,j}$, the Laplace transform of the operator is a simple polynomial in the frequency domain. The inverse operator $L_{i,j}^{-1}$ then is simply the reciprocal of $L_{i,j}$. Taking advantage of this, $L_N$ can be written in the form of a continued fraction,

$$L_N = \frac{1}{L_{1,1} + \frac{1}{L_{1,2} + \frac{1}{L_{2,1} + \ldots}} + \frac{1}{L_{2,2} + \ldots}}. \quad (3.19)$$

With $L_N$ expressed as a continued fraction a variety of techniques can be used to find either an exact or approximate convergent of the operator in the Laplace domain. Integrating back to the time domain then results in a simplified mathematical model consisting of a single ODE for the complete system.

While the expressions seen in Equations (3.18) and (3.19) are valid only for
bifurcating tree networks, they can be generalized to $P$-furcating networks with

\[
\mathcal{L}_N = \left\{ \sum_{j=1}^{P} \left\{ \mathcal{L}_{1,j} + \left( \sum_{k=1}^{P} \mathcal{L}_{2,2(j-1)+k+1} \right) + \left( \sum_{l=1}^{P} \mathcal{L}_{3,2(k-1)+l} \right) \right\}^{-1} \right\}^{-1},
\]

(3.20)

and

\[
\mathcal{L}_N = \frac{1}{\mathcal{L}_{1,1} + \frac{1}{\mathcal{L}_{2,1} + \ldots}} + \ldots + \frac{1}{\mathcal{L}_{1,P} + \frac{1}{\mathcal{L}_{2,(P-1)P+1} + \ldots}}.
\]

(3.21)

For smaller systems, $\mathcal{L}_N$ can be calculated for any $\mathcal{L}_{i,j}$’s. For large $N$, $\mathcal{L}_N$ can either be calculated in the same way, or can be approximated as $\mathcal{L}_\infty$, which, for certain forms of similarity in the operators and/or in the system’s structure, can result in a very clean and simple expression. Equations (3.18) and (3.19) are in the most general form for a bifurcating tree network, where each of the operators $L_{i,j}$ could be completely different and unrelated to one another and the only similarity that exists in the system is the result of its physical structure. However, other forms of similarity can exist with respect to the operators themselves and can often allows (3.18) and (3.19) to be reduced.

The first form of similarity considered is that within each generation. If the operators within the $i^{th}$ generation are all identical, the system is said to be symmetric. If the operators within the $i^{th}$ generation are not identical, then the system is said to be asymmetric. The second form of similarity is between the operators of successive generations. If the operators present in the $i^{th}$ generation are dependent on either the generation number $i$ or the operators in the $(i - 1)$ generation, then the system is said to be generationally dependent. If the operators in the $i^{th}$ generation are independent of the generation number and the operators of the previous generation, then the system is said to be generationally independent.
3.2.3 Continued fractions and fractional-order operators

Before proceeding it should be noted that although the previous analysis is focused on trees, the concept is applicable to a much more general class of problems where modeling results in a very large set of DAEs. For an infinite tree, the problem becomes one of reducing and solving an infinite number of equations which could be much simpler than solving a large, but finite, equation set. The physical structure of the tree provides an easy method of spotting a pattern or similarity within the equation set which allows it to be reduced or simplified. Patterns or similarity modes can also be found in other large or infinite systems of equations that would allow reduction to a single equation regardless of whether the equation set is the result of a potential driven network, chemical reaction, or mechanical system.

Furthermore, the continued fraction representation of the overall system operator in Laplace domain allows for several interesting possibilities, such as the appearance of fractional-order derivatives and integrals as will be seen. Continued fractions have been used since the time of Euclid, first as a way to write rational numbers, and later as a method to approximate irrational numbers and calculate square roots. If and only if the continued fraction representation of a number is finite, is a number rational. If the continued fraction representation of a number in infinite, then the number is irrational. Since each branch operator $L_{i,j}$, is a polynomial of some integer degree in the Laplace variable $s$ (assuming each of the operators to be linear combinations of integer-order integro-differential operations), it is no surprise that a finite continued fraction representing the overall system operator of an $N$-generational tree results in a polynomial of integer-order. However, the continued fraction representation of the operator of an infinite bifurcating tree is infinite, which will allow for polynomials in $s$ that are not integer-order, as will be seen. When these non-integer order polynomials in the Laplace domain are
transformed back to the time domain, fractional derivatives appear in the overall system operator.

3.3 Forms of similarity

The two forms of similarity considered allow bifurcating tree-systems to be classified into four distinct groups: symmetric and generation independent, asymmetric and generation independent, symmetric and generation dependent, and asymmetric and generation dependent. Each of these cases allow the continued fraction representation of the system operator to be reduced to some extent.

3.3.1 Symmetric with generation independent operators

![Figure 3.3: A generationally independent symmetric tree network composed of operator $L_a$](image)

There are several forms of similarity seen in tree-like systems which allow $L_\infty$ to be solved for exactly. The first system considered is that of an exactly self-similar tree network. In addition to its physical structure, it has two forms of
operator symmetry, both within and between generations. Consider the system shown in Figure 3.3. In this case, every branch, and thus every operator, is the same. Replacing $L_{i,j}$ in Equation (3.17) with $L_a$ gives

$$L_N = \left\{ \sum_{j=1}^{2} \left[ L_a + \left\{ \sum_{k=1}^{2} \left[ L_a + \left\{ \sum_{l=1}^{2} \left[ L_a + \ldots \right]^{-1} \right]^{-1} \right]^{-1} \right\]^{-1} \right\}^{-1} \right\}^{-1}, \quad (3.22)$$

and assuming the operator $L_a$ to be integro-differential, (3.19) becomes

$$L_N = \frac{1}{L_a + \frac{1}{L_a + \ldots} + \frac{1}{L_a + \ldots}} + \frac{1}{L_a + \frac{1}{L_a + \ldots} + \frac{1}{L_a + \ldots}}, \quad (3.23)$$

which can be rewritten as

$$L_N = L_a \left( 1 - \left( \frac{1}{2} \right)^N \right). \quad (3.24)$$

It is easy to see from Equation (3.24) that for large systems, $L_N$ can be approximated very accurately by $L_\infty$. Taking the limit $\lim_{N \to \infty} L_N$,

$$L_\infty = L_a. \quad (3.25)$$

Example: Tree of resistors

Consider a tree of the form shown in Figure 3.1 constructed of wire and identical resistors, so that each branch has one resistor of resistance $R$. In this case, the current $i(t)$ through each branch is driven by the potential difference $v(t)$ across the branch. Like the model proposed, the current is also conserved at every branching junction. For a tree network of $N$ generations, the voltage/current relationship through the entire network is given by

$$R \left( 1 - \left( \frac{1}{2} \right)^N \right) i(t) = v(t), \quad (3.26)$$

where $R$ is the resistance of one of the component resistors and $i(t)$ and $v(t)$ is the current through and the voltage difference across the entire network, respectively.
In the limit as $N \to \infty$,

$$R_i(t) = v(t).$$

(3.27)

### 3.3.2 Asymmetric with generation independent operators

Another commonly encountered structure is the asymmetric tree. In this case, a bifurcation leads to two paths, each of which are governed by a different operator. Each successive bifurcation results in the same two branches (Figure 3.4). By replacing $L_{i,j}$ with $L_1$ and $L_2$ for $j$ odd and even, respectively, Equation (3.17) becomes

$$L_N = \left\{ \sum_{j=1}^{2} \left[ L_j + \left\{ \sum_{k=1}^{2} \left[ L_k + \left\{ \sum_{l=1}^{2} \left[ L_l + \ldots \right]^{-1} \right]^{-1} \right]^{-1} \right]^{-1} \right]^{-1} \right\}^{-1},$$

(3.28)

and assuming the operators $L_1$ and $L_2$ to be integro-differential, (3.19) becomes

$$L_N = \frac{1}{L_1 + \frac{1}{L_1 + \ldots + \frac{1}{L_2 + \ldots + \frac{1}{L_1 + \ldots + L_2 + \ldots}}}}.$$  

(3.29)

For small systems, the continued fraction in (3.29) can easily be solved for $L_N$. For large scale systems $L_N$ can often be approximated by $L_\infty$, but the value of $N$ at which the system can be accurately approximated as an infinite system depends on the branch operators $L_1$ and $L_2$. For most combinations of operators, the number of generations necessary for a system to be considered effectively infinite is surprisingly small. In most cases, the behavior of a system with only $N \approx 10$ generations varies very little from that of an infinite system. For example, the convergence of $L_N$ to $L_\infty$ in the Laplace domain is shown for two different combinations of $L_1$ and $L_2$ in Figure 3.5. In both cases, $N = \infty$ is an adequate approximation for $N \approx 10$. 
With the infinite approximation, simple and compact solutions for the overall system operator can be found. In the limit as \( N \to \infty \), the continued fraction in (3.29) can be rewritten as

\[
\mathcal{L}_\infty = \frac{1}{\mathcal{L}_1 + \mathcal{L}_\infty + \mathcal{L}_2 + \mathcal{L}_\infty},
\]

which simplifies to

\[
\mathcal{L}_\infty = \sqrt{\mathcal{L}_1 \mathcal{L}_2}.
\]

Physical Example: A fractance device

A typical RLC circuit is composed of capacitors, resistors, and inductances which have impedances \( Z \propto (j\omega)^m \), where \( m \) is \(-1, 0, \) and \(+1\), respectively. A fractance is a device which does not fall into any of the three common cases, but instead has an impedance \( Z \propto (j\omega)^\alpha \), where \( \alpha \) is not an integer. Nakagawa and Soriachi [43] use an infinite tree of resistors and capacitors of the form of Figure 3.1,
Figure 3.5: $\mathcal{L}_\infty$ shown with $\mathcal{L}_N$ for different values of $N$. As $N$ increases, $\mathcal{L}_N$ converges to $\mathcal{L}_\infty$ in the Laplace domain while in the time domain $L_N$ converges to $L_\infty$. 
with $L_{i,j}$ replaced by resistors for $j = \text{odd}$ and capacitors for $j = \text{even}$, to produce such a device. For a branch with a resistor, the voltage/current relationship is given by $R_i(t) = v(t)$, where $R$ is the resistance of the resistor, $i(t)$ and $v(t)$ are the current and voltage, respectively. The branches with capacitors are governed by

$$\frac{1}{C} \int_0^t i(t) dt = \Delta v(t), \quad i(0) = 0,$$

where $C$ is the capacitance of the capacitor and $i(t)$, $v(t)$ are again, the time-dependent current and voltage, respectively. In the notation of Equation (3.1), $q = i$, $\Delta p = \Delta v$. Taking the operators to be $L_{i,j=\text{odd}} = R(\cdot)$ for the resistor branches and $L_{i,j=\text{even}} = \frac{1}{C} \int (\cdot) dt$ for the branches with capacitors, their frequency domain counterparts are given by $\mathcal{L}_{i,j=\text{odd}} = R$ and $\mathcal{L}_{i,j=\text{even}} = 1/(Cs)$. Using these operators and finding the convergent of the infinite continued fraction (3.19) gives $\mathcal{L}_\infty = (R/C)^{1/2} s^{-1/2}$. The impedance as a function of $\omega$ is found by replacing the Laplace variable $s$ with $j\omega$, giving $Z = (R/C)^{1/2}(i\omega)^{-1/2}$, which is indeed a fractance device. Converting back from the Laplace domain,

$$v(t) = \frac{(R/C)^{1/2}}{\Gamma(1/2)} \int_{-\infty}^t \frac{i(\tau)}{(t-\tau)^{1/2}} d\tau,$$

which gives $v(t)$ as exactly the $1/2$th integral of $i(t)$. Equation (3.33) can also be rewritten to show $i(t)$ as the fractional-order $1/2$ derivative of the voltage.

Finding the infinite network solution in this case is simple. More challenging is to find solutions for large, but finite, networks. In this case, a simple and easily invertible expression for the convergent of the continued fraction in Equation (3.19) is not available. For smaller networks ($N \lesssim 10$) the convergent can be found and numerically inverted, although it is increasingly difficult to do so as the size of the network increases. For networks larger than $N \approx 10$, even the numerical Laplace inversion of Equation (3.19) becomes prohibitively expensive. Figure 3.7
Figure 3.6: Step-response of a generation-dependent fractance device. ▶, ◊, ◅, □, and ◇, representing $N = 1, 2, 4, 6, \text{ and } \infty$, respectively; $E$, the $L_2$ error between the finite and infinite response for generation-dependent fractance networks of varying size.
shows the voltage response to a step input in current for several small networks, as well as the step-response of the infinite case. In this case, the step response of the infinite network is unbounded due to its infinite resistance. As steady state is never reached, any error would be somewhat arbitrary and as such is not included. But it can clearly be seen that as the size of the network increases, solutions approach that of the infinite network solution. For short times, small networks again provide an excellent approximation.

The fractance device problem becomes more difficult if the generational dependence of the pipe flow example is added. Assuming that the resistance and capacitance in generation $i$ is given as $R_i = \beta^i R$ and $C_i = C / \beta^i$, where $\beta \leq 1/2$.  

Figure 3.7: Step-response of a fractance device. $\triangleright, \diamondsuit, \triangleleft, \square$, and $\circ$, representing $N = 2, 4, 6, 8, \text{and } \infty$, respectively.
to ensure that the resistance of the infinite network remains finite, then in the frequency domain the operators are $L_a = \beta^i R$, and $L_b = \beta^i / (Cs)$. For the infinite network, substituting into Equation (3.19) gives a quadratic equation in $L_\infty$,

$$(2\beta - \beta^2)L_\infty^2 - (\beta - 1)(L_a + L_b)L_\infty + L_a L_b = 0, \quad (3.34)$$

resulting in two possible solutions. Numerically inverting back to the time domain, both solutions can be found. In this case, the solution resulting from the negative root can be seen to be spurious as the solution does not behave as would be intuitively expected.

Figure 3.6 shows the step-response of the generationally-dependent fractance device using the positive root of $L_\infty$. As before, solving the infinite network case is simpler than solving for large finite networks. Assuming large networks to be infinite, however, again provides an acceptable approximate solution. Figure 3.6 shows the step-response of $N = 1, 2, 6, 8$ and $\infty$ generation networks, and the error between the infinite and finite networks with increasing generation number $N$. The error is computed in the same manner as (3.49).

Example: Fractional-order viscoelasticity models

There exist many different mathematical models to describe viscoelastic behavior, most of which are a combination of the classic models that have been used to describe elastic and viscous materials for years. The classical models use simple mechanical elements such as springs and dampers to model the elastic and viscous behavior, respectively, of such materials, but neither model correctly describes viscoelasticity. Early models attempted to capture viscoelastic behavior by combining, in series or parallel, the individual mechanical elements used to describe elastic and viscous materials. For example, a spring models elastic behavior with
stress being purely a function of the instantaneous strain, \( \sigma = k \epsilon \). A damper models viscous behavior with stress dependent only on the instantaneous strain rate, as \( \sigma = \mu \frac{d \epsilon}{dt} \). The classical Voigt model seeks to combine the elastic and viscous behavior seen in viscoelastic materials by combining a spring and a damper in parallel (Figure 3.8). The Voigt model, while it does a much better job of describing viscoelastic materials than the traditional spring or damper models, still does not capture true viscoelastic behavior. It has been suggested by Torvik and Bagley [59] and Heymans and Bauwens [30] that true viscoelastic materials behave not as a combination of a spring, \( \sigma \propto \frac{d \epsilon}{dt} \), and damper, \( \sigma \propto \frac{d^2 \epsilon}{dt^2} \), but rather as something in between, \( \sigma \propto \frac{d^\alpha \epsilon}{dt^\alpha} \), where \( 0 < \alpha < 1 \).

Heymans and Bauwens construct such a model by using an asymmetric, generation independent infinite tree network composed of the classical spring and damper elements as shown in Figure 3.9. In this case, the equations for each branch are given by

\[
\begin{align*}
  k \epsilon &= \sigma, \\
  \mu \frac{d \epsilon}{dt} &= \sigma.
\end{align*}
\]  

(3.35)

In the notation of (3.1), \( q = \sigma, \Delta p = \epsilon \), creating some difficulty in writing \( L_1 \) and
\( L_2 \). However, in the Laplace domain, \( L_1 = 1/k \) and \( L_2 = 1/\mu s \), so from (3.31),

\[
L = \sqrt{\frac{1}{k\mu s}}.
\] (3.36)

Transforming back from the Laplace domain,

\[
L = \sqrt{\frac{1}{k\mu} \frac{d^{-1/2}}{dt^{-1/2}}},
\] (3.37)

giving the fractional-order stress-strain relationship

\[
\sqrt{k\mu \epsilon} = \frac{d^{-1/2} \sigma}{dt^{-1/2}},
\] (3.38)

or strain as the one-half integral of the stress. Alternatively, and more in line with traditional appearance, the stress can be written as a function of strain,

\[
\sigma = \sqrt{k\mu} \frac{d^{1/2} \epsilon}{dt^{1/2}},
\] (3.39)

giving stress as the one-half derivative of the strain. This model, while still not a perfect description of viscoelastic behavior, performs much better than the classic descriptors. Its worth mentioning that other researchers have developed fractional-order viscoelastic models using completely different methods. One of the most interesting of these models is developed from a statistical approach where the fractional-order derivatives appear to randomness [10].

3.3.3 Symmetric with generation dependent operators

Perhaps the most common form of similarity seen in tree-like systems is the symmetric, generation dependent pattern seen in Figure 3.10. In this system the operators \( L_{i,j} \) depend only on the generation, \( i \), in which the operator resides. Generational dependence is often a direct result of the change of physical variables, such as branch diameter or length, from generation to generation. Many biological systems such as the arterial and bronchial networks could be described in this
manner. As the generation number increases, the branches become smaller and smaller.

From Equation (3.17), for a symmetric, generation dependent tree,

\[
L_N = \left\{ \sum_{j=1}^{2} \left[ L_1 + \left\{ \sum_{k=1}^{2} \left[ L_2 + \left\{ \sum_{l=1}^{2} \left[ L_3 + \ldots \right]^{-1} \right]^{-1} \right]^{-1} \right\}^{-1} \right]^{-1} \right\}^{-1},
\]

(3.40)

and assuming the operators \( L_i \) to be integro-differential, (3.19) becomes

\[
\mathcal{L}_N = \frac{1}{\mathcal{L}_1 + \frac{1}{\mathcal{L}_2 + \ldots + \mathcal{L}_2 + \ldots} + \frac{1}{\mathcal{L}_1 + \frac{1}{\mathcal{L}_2 + \ldots + \mathcal{L}_2 + \ldots}}}
\]

(3.41)
which can also be written as the sum,

\[ \mathcal{L}_N = \sum_{i=1}^{n} \frac{1}{2^i} \mathcal{L}_i. \]  

(3.42)

| \( i = 1 \) | \( i = 2 \) | \( i = 3 \) | \( \cdots \) | \( i = N \) |
|---|---|---|---|

\[ \begin{array}{cccc}
L_1 & L_2 & L_3 & \cdots \\
L_2 & L_3 & \cdots & L_i \\
L_3 & \cdots & \cdots & \cdots \\
\end{array} \]

Figure 3.10: A generationally dependent symmetric tree network with operators \( L_i \)

Physical Example: Laminar pipe flow in large, self-similar trees

One such example is that of a piping network arranged in a bifurcating tree. Consider an \( N \)-generational bifurcating tree network of circular pipes transporting fluid due to a potential difference across it. If the diameter and length of the pipe are taken to vary with generation number \( i \) according to \( D_{i,j}^* = D^*/\beta^{i-1} \) and \( \ell_{i,j}^* = \ell^*/\beta^{i-1} \) where \( D^* = D_{1,j}^* \) and \( \ell^* = \ell_{1,j}^* \), where \( * \) denotes a dimensional quantity, then appropriate values of \( \beta \) can be chosen to ensure that all physical quantities, such as length and volume, remain finite even for an infinite system [22].
Using a one-dimensional flow model for laminar pipe flow which can be derived from the Navier-Stokes equations and is used by Franco [21], the dimensional momentum equation for a branch is

\[
\frac{dq_{i,j}^*}{dt^*} + \frac{32\nu^*}{D_{i,j}^*} q_{i,j}^* = \frac{\pi D_{i,j}^*}{4\rho^*\ell_{i,j}^*} \Delta p_{i,j}^*.
\] (3.43)

Where \( t^* \) is time, \( q^* \) the flow rate, \( \Delta p^* \) is the pressure drop across a pipe, \( D^* \) is the diameter and \( \ell^* \) the length of pipe, and \( \nu^* \) and \( \rho^* \) are the kinematic viscosity and density of the fluid, respectively. Nondimensionalizing by \( q_{i,j} = q_{i,j}^*/(\nu^* D_{i,j}^*) \), \( p_{i,j} = \pi D_{i,j}^3 p_{i,j}^*/(128\nu^*\rho^*\ell^*) \), and \( t = 32\nu^*/D^* \), and substituting for \( D_{i,j}^* \) and \( \ell_{i,j}^* \), Equation (3.43) becomes

\[
\beta^{i-1}\frac{dq_{i,j}}{dt} + \beta^{3(i-1)} q_{i,j} = \Delta p_{i,j}.
\] (3.44)

In the form of \( L_{i,j} q_{i,j} = \Delta p_{i,j} \), the generational dependent operator can be written as \( L_{i,j} = \beta^{i-1} d/dt + \beta^{3(i-1)} \), and in the Laplace domain, \( \mathcal{L}_{i,j} = \beta^{i-1}s + \beta^{3(i-1)} \).

Using Equation (3.19) and solving for the convergent to the continued fraction in an \( N^{th} \)-generation network,

\[
\mathcal{L}_N = \sum_{i=1}^{n} \frac{1}{2^i} \{ \beta^{i-1}s + \beta^{3(i-1)} \},
\] (3.45)

\[
= \frac{1}{2} \left[ \frac{1 - (\beta/2)^N}{1 - \beta/2} s + \frac{1 - (\beta^3/2)^N}{1 - \beta^3/2} \right],
\] (3.46)

which is a simple polynomial in the frequency variable \( s \) that is dependent on \( N \).

Transforming back from the Laplace domain an exact analytical solution for the behavior of the entire network can be found,

\[
L_N = \frac{1}{2} \left[ \frac{1 - (\beta/2)^N}{1 - \beta/2} \frac{d}{dt} + \frac{1 - (\beta^3/2)^N}{1 - \beta^3/2} \right].
\] (3.47)

In this example a large set of DAEs is reduced to a simple ODE, and an exact solution for a network of any size \( N \) can be found easily. But it is often the case that for very large \( N \), finding a simple analytical expression such as Equation
(3.46) is not trivial or may not even be possible. Even if an analytical expression can be found, transforming back to the time domain may be challenging. While simple and exact solutions may not be possible, approximate solutions can be easily found. From Equation (3.47), as \( N \) tends to infinity, solutions converge to the infinite network case. In this case, for \( 1 < \beta < 2^{1/3} \) (the same restrictions on \( \beta \) can be shown to be required for all network quantities to remain finite) the solution converges to

\[
L_\infty = \frac{1}{2} \left[ \frac{1}{1 - \beta/2} \frac{d}{dt} + \frac{1}{1 - \beta^3/2} \right].
\]  

(3.48)

Figure 3.11 shows the step response of bifurcating piping networks for different values of \( N \) and the error between those responses and that of the infinite system. In this case, the error, \( E \), is defined as

\[
E = \int_{t_{\text{min}}}^{t_{\text{max}}} \left\{ q_\infty(t) - q_N(t) \right\}^2 dt.
\]  

(3.49)

This definition is chosen to best represent the difference between the dynamic responses of the finite and infinite solutions. It is obvious from Figure 3.11 that as the size of the system increases, solutions converge to the infinite system response and that even relatively small systems can be well approximated by an infinite network. In this case, approximating the network as infinite provides a very good approximate solution even for networks of only \( N = 8 \) or \( N = 12 \) generations.

3.3.4 Asymmetric with generation dependent operators

The asymmetric with generation dependent operator case could be used to describe any potential driven bifurcating tree network, and the previous three cases, in fact, are all subsets of this set. This most general case is represented by Equations (3.17) and (3.19) and without any form of similarity, either within each generation (asymmetric) or between generations (generation dependent), no
Figure 3.11: Step-response of bifurcating tree pipe networks. ⃝, □, ⋄, △, and solid line representing $N = 2, 4, 8, 12,$ and $\infty$, respectively; $E$, the $L_2$ error between the finite and infinite response for fractance networks of varying size.
general simplification is possible. It is, however, possible to make further simplification under certain circumstances and for special cases. For example, consider the asymmetric, generation dependent tree shown in Figure 3.12. In this example, a relationship between the operators in different generations allows for some minor simplification. The operators \( L_{i,j} \) can be written \( L_{i,j} = \beta^{i-1}L_a \) and \( L_{i,j} = \beta^{i-1}L_b \) for \( j = \text{odd} \) and \( \text{even} \), respectively.

\[
\begin{array}{c|c|c|c|c}
 i = 1 & i = 2 & i = 3 & \cdots & i = N \\
 p_{in} & L_a & \beta L_a & \beta^2 L_a & \beta^i L_a \\
 & L_b & \beta L_b & \beta^2 L_b & \cdots \\
 & & \beta L_a & \beta^2 L_a & \beta^i L_a \\
 & & L_b & \beta L_b & \cdots \\
 & & & \beta L_a & \cdots \\
 & & & \beta L_b & \cdots \\
 & & & \cdots & \beta^i L_b \\
\end{array}
\]

Figure 3.12: A generationally dependent asymmetric tree network.

With the operators shown in Figure 3.12, Equation (3.19) can be written

\[
\mathcal{L}_N = \frac{1}{\mathcal{L}_a + \frac{1}{\beta \mathcal{L}_a + \ldots} + \frac{1}{\beta \mathcal{L}_b + \ldots}} + \frac{1}{\mathcal{L}_b + \frac{1}{\beta \mathcal{L}_a + \ldots} + \frac{1}{\beta \mathcal{L}_b + \ldots}}. \quad (3.50)
\]

For small \( N \), this can be solved for \( \mathcal{L}_N \). For larger systems, or as \( N \to \infty \), this
can be approximated as

\[ L_\infty = \frac{1}{L_a + \beta L_\infty + \frac{1}{L_b + \beta L_\infty}}, \tag{3.51} \]

which provides a quadratic equation in \( L_\infty \) which can then be solved.

Physical example: Steady state conduction through an infinite bifurcating tree

Consider one-dimensional steady state conduction through an infinite, fractal-like tree network of connected cylindrical bars removing heat from a point source. The heat transfer in each bar is governed by \( q = \frac{kA}{\ell} \Delta T \), where \( q \) is the heat transfer rate, \( \Delta T \) is the temperature difference across each bar, \( k \) is the thermal conductivity of the bar, and \( \ell \) and \( A \) are the length and cross sectional area of the bar, respectively. If the bars are physically arranged as shown in Figure 3.13 where \( \gamma \) is the ratio between the lengths of the bars at each branching point and \( \alpha \) is the ratio between the total length and cross sectional areas of bars in successive generations, then \( \ell_{a,i} = \ell_a/\alpha^{i-1} \) and \( \ell_{b,i} = \gamma \ell_a/\alpha^{i-1} \), and \( r_{a,i} = r_{b,i} = r/\alpha^{i-1} \). In the notation of Figure 3.12, this gives

\[ L_a = \frac{\ell_a}{kA_a} \tag{3.52} \]
\[ = \frac{\ell_a}{k\pi r^2} \tag{3.53} \]
\[ L_b = \frac{\ell_b}{kA_b} \tag{3.54} \]
\[ = \frac{\gamma \ell_a}{k\pi r^2} \tag{3.55} \]

and

\[ \beta = \alpha. \tag{3.56} \]

Now, from Equation (3.51),

\[ L_\infty = \frac{1}{L_a + \alpha L_\infty + \frac{1}{L_b + \alpha L_\infty}}, \tag{3.57} \]
which gives

\[ \mathcal{L}_\infty = \frac{1 - \alpha - \gamma - \alpha \gamma + \sqrt{4(2\alpha - \alpha^2)\gamma + 1 - \alpha - \gamma - \alpha \gamma}}{2(-2\alpha + \alpha^2)} \frac{\ell_a}{k\pi r^2}, \]  

(3.58)

where the spurious root of (3.57) has been discarded. In the time domain, \( L_\infty \) and \( \mathcal{L}_\infty \) are the same as the problem is steady state, so the total heat removed is

\[ q_{tot} = \frac{2(-2\alpha + \alpha^2)}{1 - \alpha - \gamma - \alpha \gamma + \sqrt{4(2\alpha - \alpha^2)\gamma + 1 - \alpha - \gamma - \alpha \gamma}} \frac{k\pi r^2}{\alpha} \Delta T. \]  

(3.59)

Figure 3.13: Fractal-like network of conducting rods for heat removal from a point source.

3.4 Conclusions: The forest through the trees

The use of self-similar networks, such as the bifurcating tree, is commonplace in science today as a model for a variety of naturally occurring and artificial systems. A traditional reductionist approach to analysis of such systems, however, can be difficult. Even for relatively small networks, dynamic solutions can be difficult to obtain due to the size of the ensuing system of coupled DAEs. For very large
networks, solutions can become effectively intractable making model simplification or reduction a necessity. But systems with a physically self-similar structure can allow the complexity of mathematical models to be reduced, often to a single simple equation describing the input/output behavior of the entire system. For systems composed of components governed by integro-differential operators, the overall system transfer function can be written in the form of a continued fraction. Further forms of similarity within the component operators, and thus in the system of DAEs, can lead to further simplification of the transfer function and often to very simple and compact solutions in the time domain.

In the case of bifurcating tree networks, the self-similar structure of the system and its resulting mathematical model allows for massive reduction. It can be expressed in the frequency domain in terms of a continued fraction. And by finding the convergent of the fraction the large or even infinite system of DAEs from the original mathematical model can be reduced to a single ODE.

As was shown in the included examples, it may only take a relatively small number of generations before the response rate approaches that of the infinite network. In light of these examples, the question should then be asked why a very large system should be approximated as infinite when even approximating the system with as few as \( N = 6 \) generations could yield an acceptable solution. This method—reduction of complexity by a reduction in size—is indeed the approach taken most often when modeling large systems. But it should be noted that the systems shown in this chapter are exceptional cases chosen precisely because of the fact that they allow for simple solutions of smaller networks. Generally speaking, this is not the case. Consider that even an \( N = 6 \) generation results in an order-2 DAE system of 188 coupled equations. For more complicated system operators, this is not trivial at all. And in these cases reduction of complexity by approxi-
mating the system as infinite then has the advantage of producing both a more tractable and more accurate solution.

The continued fraction representation of the overall system operator also offers insight into how the behavior of a tree network is affected by increasing size. Most notably, as the number of generations tends towards infinity, a network composed entirely of integer-order components has the ability to behave as a fractional-order system. While this change in behavior only occurs in the impractical infinite limit, it is by no means a useless observation. In fact, the infinite limit is actually quite useful and very practical in real world networks, as the continued fraction representation of the overall system operator can show that networks consisting of as little as $N \approx 10$ generations are hardly affected by the addition of subsequent generations.

Finally, while the development in this chapter focused on linear systems with a perfect binary tree structure, it should be noted that these requirements are not so restrictive as to make it inapplicable to real-world systems. A similar approach can be taken to analyze systems with other self-similar or nearly self-similar structures. While the requirement that the systems be linear is a strong one, it is still helpful for large scale non-linear systems. In these cases, linearization is just another step in the reduction of a large scale, non-linear set of DAEs to a simpler mathematical model. Although, for systems with an extreme sensitivity to non-linearities, other methods must of course be pursued.

In the following chapter, we will consider networks with non-linear operators and a method of reduction that can be used without linearizing away those non-linearies.
This chapter presents two numerical projection methods for dynamic analysis of large-scale flow networks. For the interested reader, the content presented in this chapter is available in [41].

In the previous chapter, we considered large scale potential-driven transfer networks in the form of self-similar bifurcating trees. In the particular cases considered, we were able to take advantage of the self-similarity in the mathematical equations resulting from the networks self-similar structure to solve for a much simpler and reduced mathematical model. In the end, an infinite system of equations were reduced to a single equation describing the over all input-output.

The usefulness of that reduction, however, depended on the ability to calculate the inverse operators of each individual component operators. But with the assumption that we were dealing with a system of linear equations, it was shown that further simplification could be made in Laplace space and the system operator could be represented as a continued fraction in the Laplace variable $s$. This then allowed us to find a very simplified (either exact or an excellent approximation) system operator which could then be converted back to the time domain.

The useful reduction achieved in the previous chapter then was really dependent on the condition that all operators being considered are linear. But in many
cases the linear assumption may be too strong for system being considered. As an example, consider again the self-similar bifurcating piping network analyzed in Section 3.3.3 which assumed that the flow through each component pipe was laminar in nature, thus giving a simple first-order linear ODE model for each branch. If the laminar flow assumption was relaxed such that flow could be either laminar or turbulent depending on the Reynolds number of the flow in the pipe, then the problem becomes much more complicated. Not only does the DAE system become non-linear, but the system of equations (the mathematical model) itself can now vary in time as laminar-turbulent transition can occur in each branch individually.

In this chapter, the self-similar bifurcating piping network is considered again as the motivating example, but the assumption of linearity on the resulting DAE system has been removed. As reduction utilizing a continued fraction in the Laplace domain cannot be used due to the non-linear equations set, the analytical reduction method of Chapter 3 to provide dynamic solutions is inapplicable. Instead, dynamic solutions must be developed numerically. As pipe flow is a particular case of the Navier-Stokes equations governing all fluid flow, and much research has been conducted to numerical solutions to this equation set, there is a wealth of methods available that could be applied. In the following sections, one particularly famous solution method, Chorin’s Method, will be used and extended to provide a simplified solution method for the DAE system resulting from self-similar bifurcating piping networks, and more generally, other piping networks.

4.1 Existing methods for analysis of piping networks

Methods for steady-state analysis of complex piping networks have existed for many years, but dynamic analysis of the same can be difficult – particularly for
large scale or high-dimensional systems. The first procedure to gain regular use for steady-state analysis of piping networks was proposed by Cross [15]. This uses an iterative algorithm to determine the flow rates and nodal pressures in a network by incrementally changing the flow through each pipe in every closed loop by the same amount until the solution converges. The first flow rate guess satisfies mass conservation at each junction, and pressure corrections at each step are done in such a way to maintain that status. The usefulness of Hardy-Cross was first and foremost in its simplicity; it could be computed by hand in a spreadsheet manner. In the following years, as networks continued to grow in size and complexity, software implementations of this and other steady-state approaches came to be commonly used [36, 44].

While steady-state analysis of piping networks involves solving a set of nonlinear algebraic equations, dynamic analysis requires the solution of a system of nonlinear differential-algebraic equations (DAEs). Solving DAEs numerically has its own difficulties as it requires the simultaneous solution of a system of ODEs and a corresponding set of separate algebraic constraints [46], but adding nonlinearity to the ODEs only makes it more complex. There is a wealth of literature on the topic of solving general DAEs [3, 4, 8, 25, 26, 37, 49, 50] which is applicable to the dynamic analysis of piping networks, but none geared to take advantage of the specific nature of the DAEs resulting from a flow distribution network. A few of the more common and most clever approaches used to solve DAEs include the differentiation index approach [8] that simplifies the system by differentiating until pure ODEs can be realized, projector-based methods that seek to obtain an inherent ODE that governs the system by projecting the original set into certain characteristic subspaces [37], and reduction techniques [49] that can be roughly described as eliminating certain variables by solving the constraints [50].
Under certain assumptions, such as laminar flow, the DAEs governing flow in a network are linear. In this case techniques exist that can be employed to obtain solutions. For specific cases such as self-similar tree networks, the DAEs can be reduced to a single linear ODE and exact analytical solutions can be found [22, 40]. For dynamic analysis of a generic piping network, even if laminar flow is assumed, the solution of DAEs is required.

While all general DAE solution procedures can be applied to one-dimensional flow in a piping network, none is tailored to its use. There are, however, a number of computational techniques that have been developed for the fundamental equations in fluid mechanics that can be adapted. Of particular interest are those developed for the solution of the Navier-Stokes equations that govern fluid motion, the discretization of which often results in nonlinear DAEs. One that will lend itself well to adaptation to piping networks is Chorin’s projection method [12] (referred to simply as Chorin’s method hereafter) in which the PDEs arising from the momentum equation are integrated and the solution is then projected onto a solution space where the algebraic mass conservation constraint is valid. Strictly-speaking, it is not a method to solve generic DAEs like others [3, 4, 8, 25, 26, 37, 49, 50]. Instead, it solves the DAEs by taking advantage of the special relationship between the differential momentum equations and the algebraic constraints provided by conservation of mass. As the DAEs resulting from one-dimensional flow in a piping network are related to the discretized Navier-Stokes equations, the latter provide an excellent framework for the former.

In the following sections, Chorin’s method is discussed in more detail and extended to the specific application of piping networks. To demonstrate its accuracy, a large, but relatively simple network is presented and analyzed. The network dynamics, as computed numerically, are then compared to a case where an exact
analytical solution can be obtained. The reader should take care to note that while the following analysis is performed with piping networks in mind, the study is not so narrow in scope that it is restricted only to that, but there are other applications for which the technique can be applied equally well.

4.2 Projection methods

Chorin’s projection method for the Navier-Stokes equations is usually represented using a fractional step. The time is first discretized and an ‘intermediate’ velocity is obtained by solving the momentum equation, disregarding the divergence-free mass conservation condition and often the pressure. Then a projection onto a divergence-free space is performed and the updated pressure is found by solving a pressure-Poisson equation. Finally, the now divergence-free velocity is updated accordingly. The advantage of this numerical scheme is that it effectively decouples the computations of the velocity and pressure fields.

As motivation for the usefulness of projection in solving the time dependent incompressible Navier-Stokes equations, consider the following: after spatial discretization, the equations governing incompressible fluid flow can be written in the general form of a Hessenberg Index-2 DAE system [4] as

\[ u' = f(u, p), \quad (4.1) \]
\[ 0 = g(u), \quad (4.2) \]

where Equations (4.1) and (4.2) are the coupled momentum and conservation of mass equations, respectively, \( u \) and \( p \) are the respective column vectors resulting from the spatial discretization of the velocity \( \vec{u} \) and pressure \( p \) fields, and the prime indicates a derivative with respect to time. From a practical point of view, the essential difficulty in solving this system is caused by the lack of a differential
equation for the pressure. It can be seen by writing these equations in matrix form,

$$\mathbf{M} \begin{pmatrix} u' \\ p' \end{pmatrix} = \begin{pmatrix} f(u, p) \\ g(u) \end{pmatrix},$$

(4.3)

that if $\mathbf{M}$ were nonsingular, any numerical ODE solver could be used. Without a pressure evolution equation, however, $\mathbf{M}$ is singular requiring the use of more complicated algorithms or methods. Artificial compressibility is one such method that can serve to make $\mathbf{M}$ nonsingular. Another common technique for this is index reduction — where the algebraic constraint is differentiated until a set of ODEs is obtained. However, as the pressure is not present in the algebraic constraint, index reduction is not trivial as it takes two differentiations, and corresponding initial and boundary conditions, to produce ODEs. This is why, by definition, the system is considered of index-2.

Chorin’s scheme takes advantage of the fact that the algebraic constraints present in the DAEs — the conservation of mass — can be posed as a divergence-free condition on the velocity. And in fact, in incompressible flow $\nabla p$ simply serves as a Lagrange multiplier to make the velocity divergence-free. By then requiring that the velocity at each new time-step be divergence-free, the algebraic constraints can be ignored in the integration. The purpose here is to present an extension to the case of piping networks and, as a special case of that, to tree networks.

4.3 Problem formulation

The governing equations for one-dimensional flow in a piping network will be first developed. To this end, a self-similar piping network in the form of a tree is considered. While any network could be chosen, this specific configuration has the following advantages. First and foremost, in some cases an exact analytical solution
is available to which numerical solutions can be compared. Second, the bifurcating
tree is a common network used to model many engineering systems, both natural
and artificial, including everything from biological transport networks and river
basin drainages to viscoelasticity [1, 11, 13, 16, 19, 20, 23, 30, 38, 45, 56, 61].

4.3.1 Tree network topology

The piping network being considered is shown schematically in Figure 4.1. Each
branch in a given generation bifurcates into two in the following generation; the
tree consists of $N$ total generations. If $i$ indicates the generation number of a
branch and $j$ the branch number within that generation, then each branch can be
represented by a unique pair of integers $(i, j)$ where $1 \leq i \leq N$ and $1 \leq j \leq 2^{i-1}$,
the number of branches in generation $i$. The junctions are also uniquely identified
by the indices $(i, j)$ of the branch that leads toward it. The flow rate through any
branch is also identified using the identifiers of the branch, i.e., $q_{ij}$. The pressure $p_{ij}$
at any junction is identified by the corresponding junction $(i, j)$, while a pressure
difference $\Delta p_{ij} = p_{i-1,\lceil j/2 \rceil} - p_{ij}$, where $\lceil \cdot \rceil$ is the ceiling function, identifies the
pressure drop across branch $(i, j)$. In a tree of $N$ total generations, there are a total
of $2^N - 1$ branches and $2^{N-1} - 1$ junctions. In computational codes, however, it is
convenient to use a single running index to denote the branches and the junctions.

The inlet to the tree is a pipe of diameter $D^*$, length $L^*$, and volumetric flow
rate $q^*(t^*); t^*$ is time and $*$ denotes a dimensional quantity. Each branch $(i, j)$
is a circular pipe of diameter $D_{ij}^*$ and length $L_{ij}^*$ that carries a volume flow rate
$q_{ij}^*$. It bifurcates at junction $(i, j)$ into two branches, each of diameter $D_{ij}^*/\gamma$ and
length $L_{ij}^*/\gamma$, where $\gamma$ is the size ratio between pipes in successive generations.
With $D_{11}^* = D^*$ and $L_{11}^* = L^*$, the diameter and length of any other branch $(i, j)$
is given by $D_{ij}^* = D^*/\gamma^{i-1}$ and $L_{ij}^* = L^*/\gamma^{i-1}$. For simplicity, it is assumed that
both the length and diameter change by the same factor $\gamma$. While there is no geometrical restrictions on $\gamma$, it is assumed to be constant for the entire tree and greater than unity in value. However, there are values of $\gamma$ that have been shown to optimize certain flow parameters. In physiology, Murray’s law yields for laminar and turbulent flows, respectively, that diameter ratios of $2^{1/3}$ and $2^{3/7}$ and length ratios of $2^{1/3}$ and $2^{1/7}$ are the optimal ratios to minimize the energy needed to maintain blood flow through a bifurcating network [42].

4.3.2 Governing equations

A one-dimensional flow model is used to model the hydrodynamics of the network. It is assumed that the flow is fully developed, head loss is governed by the Darcy-Weisbach equation, and inlet and exit effects are negligible. For a single
branch \((i, j)\), the momentum equation is

\[
\frac{dq_{ij}^*}{dt^*} = -f_{ij} \frac{2}{\pi D_{ij}^*} q_{ij}^* + \frac{\pi D_{ij}^*}{4 \rho^* L_{ij}^*} \Delta p_{ij}^*,
\]

(4.4)

where \(f_{ij}\) is the friction factor for the branch, \(\rho^*\) is the fluid density, and \(\Delta p_{ij}^*\) is the driving pressure difference for that branch. The three terms in Equation (4.4) are, from left to right, the fluid acceleration, frictional resistance, and the pressure force, respectively. Strictly speaking, one should write \(|q_{ij}^*|\) in the frictional force instead of \(q_{ij}^*\), but that will not be necessary for a tree where the flow directions are all the same and pre-determined. The frictional term is dependent on the friction factor, \(f_{ij}\), which is a function of the pipe Reynolds number \(Re_{ij} = 4q_{ij}^*/\pi D_{ij}^* \nu^*\).

For laminar flow, the friction factor is \(f_{ij} = 64/Re_{ij}\) where \(Re_{ij} < Re_c\), \(Re_c\) being the critical Reynolds number for transition. For turbulent flow there are many approximations available. One can use the Blasius formula for a smooth pipe, \(f_{ij} = 0.316 Re_{ij}^{7/4}\) or assume that \(f_{ij}\) is constant for high \(Re\), among the many possibilities.

For an incompressible fluid, mass conservation at each junction \((i, j)\) can be written as

\[
q_{ij}^* - q_{i+1,2j-1}^* - q_{i+1,2j}^* = 0.
\]

(4.5)

Nondimensionalizing the time, flow rates, and pressure using \(t^* = t^* 32 \nu / D^2\), \(q_{ij} = q_{ij}^*/\nu^* D^*\) and \(p_{ij} = p_{ij}^*/\nu D^3 / 128 \nu^2 \rho^* L^*\), respectively, Equations (4.4) and (4.5) become

\[
\frac{dq_{ij}}{dt} = -F_{ij} + \beta_{ij} \Delta p_{ij},
\]

(4.6a)

\[
q_{ij} - q_{i+1,2j-1} - q_{i+1,2j} = 0,
\]

(4.6b)
where $\beta = \gamma^{-(i-1)}$ and

$$F_{ij}(q_{ij}) = \begin{cases} \\ \\ \gamma^{2(i-1)}q_{ij} & \text{laminar flow,} \\ f\gamma^{3(i-1)}q_{ij}^2 & \text{turbulent flow with constant friction factor,} \\ 0.005926\gamma^{11(i-1)/4}q_{ij}^{7/4} & \text{turbulent flow with Blasius formula.} \end{cases}$$

(4.7a)

The initial conditions $p_{ij}(0)$ and $q_{ij}(0)$ must be prescribed. The flow may be driven either by a known pressure difference between inlet and outlet, or by a known flow rate. For the latter, the flow rate $q_{ij}(t)$ may simply be substituted into Equation (4.6a) to give the pressure distribution, so this will not be considered any further. On the other hand, for pressure-driven flows

$$p_{01}(t) = p_{out} + \Delta p(t),$$

(4.8a)

$$p_{N,j}(t) = p_{out}(t), \ j = 1, \ldots, 2^{N-1}.$$  

(4.8b)

must be prescribed, where $\Delta p(t) > 0$ is the overall pressure difference that is driving the flow.

While Equations (4.6a) and (4.6b) were developed for the specific case of a bifurcating tree network, it must be noted that any piping network consists of equations of the same form: the flow through each branch is governed by a momentum equation of the form (4.4) and a mass conservation equation of the form (4.5) exists for each internal junction.

4.4 Projection methods for network equations

In this section, two numerical projection methods are presented for the modeling of flows in nonlinear piping networks. First, the system of the governing equations is transformed to a matrix form suitable for analysis. The key is that
the internal incidence matrix of the network and its transpose coincide with the matrices of pressure difference and the mass conservation constraint, respectively. Second, the projection method outlined above in Section 4.2 is extended to piping networks, and first- and second-order approximations are derived and validated for consistency and stability.

4.4.1 Matrix form of equations

An incidence matrix can be written for any directed graph or network to describe the connections between every node and branch. For a tree network, an incidence matrix, \( A = [a_{ij}] \), is defined such that the element \( a_{ij} \) is 1 if branch \( i \) goes into junction \( j \), \(-1\) if it goes out of, and 0 otherwise. An \( N \)-generation piping network then produces a \((2^N - 1) \times (2^N - 1 - 1)\) incidence matrix. It is also necessary to define a \((2^N - 1) \times 1\) vector of flow rates in the branches \( q = [q_{11}, q_{21}, q_{22}, q_{31}, \ldots, q_{N,2^N-1}]^T \), and a \((2^N - 1) \times 1\) vector of internal pressures at the junctions \( p = [p_{11}, p_{21}, p_{22}, p_{31}, \ldots, p_{N-1,2^N-2}]^T \).

With these definitions, Equations (4.6a) and (4.6b) can be written in matrix form

\[
\frac{dq}{dt} = -F - M A p + M p_{i/o}, \quad (4.9a)
\]

\[
A^T q = 0. \quad (4.9b)
\]

where \( F(q) \) stands for frictional terms which may be nonlinear, \( M \) is a diagonal matrix with \( \beta_{ij} \) being its elements on the main diagonal, and \( p_{i/o}(t) = [\Delta p(t), 0, \ldots, 0]^T \) is the known time-dependent pressure difference that drives the flow. As mentioned before, it is important to note that the incidence matrix \( A \) appears in both the pressure term in Equation (4.9a) and the mass conservation constraint of Equation (4.9b). This is of great significance in extending Chorin’s
method to piping networks. It is also worth noting that, in general, the incidence matrix $A$ is very sparse – even for well-connected networks. Only in the case of extremely dense networks will this not be true.

4.4.2 Numerical algorithm

Using very similar steps, Chorin’s method can be extended to piping networks because of the specific nature of the DAEs to be solved. Using initial conditions and the overall driving pressure difference, the scheme is implemented numerically to advance the solution for a time step of $\Delta t$. Thus, from a known flow rate $q^n$ that satisfies Equation (4.9b), the flow rate $q^{n+1}$ and pressure distribution $p^{n+1}$ that satisfy Equations (4.9) is to be determined.

The new vector of flow rates is given by the semi-implicit equation

$$\frac{q^{n+1} - q^n}{\Delta t} = -F - MA p^{n+1} + M_{i/o} p_{i/o}^{n+1},$$

(4.10)

where superscripts denote the time step at which the vector is evaluated and $F$ is the frictional term, which will be discussed later but is irrelevant to the generalities of the projection scheme.

From Equation (4.10), the pressure at time $n + 1$ is needed before the vector of flow rates can be updated. Toward this end, an intermediate flow rate vector is first calculated by

$$\frac{q^* - q^n}{\Delta t} = -F - MA p^n + M_{i/o} p_{i/o}^{n},$$

(4.11)

where $q^*$ differs from $q^{n+1}$ in that the mass conservation equation has been disregarded and the pressure at time $n$ has been used. Since there is not an evolution equation for the pressure, it is assumed that the pressure at $n + 1$ can be decomposed into

$$p^{n+1} = p^n + \phi,$$

(4.12)
the sum of the pressure at the current time step and a pressure correction \( \phi \).

Substituting Equation (4.12) into (4.10) and subtracting (4.11), the pressure correction step relating \( q^* \) and \( q^{n+1} \) is

\[
\frac{q^{n+1} - q^*}{\Delta t} = -M A \phi + M (p^i_{i/o}^{n+1} - p^i_{i/o}^n).
\]

To solve for \( \phi \), recall that \( A^T q^{n+1} = 0 \) must be enforced. Left-multiplying Equation (4.13) by \( A^T \) and solving for \( \phi \) gives

\[
\phi = \frac{1}{\Delta t} (A^T M A)^{-1} A^T q^* + (A^T M A)^{-1} A^T M (p^i_{i/o}^{n+1} - p^i_{i/o}^n),
\]

effectively making \( \phi \) the pressure correction that projects \( q^{n+1} \) onto the manifold of vectors for which the mass conservation restraint is valid. It should be noted that, in general, \( A^T M A \) is singular, so it must be regularized to compute the inverse. But with \( \phi \) now known, the vector of flow rates can be updated using Equation (4.10) or (4.13). The pressure can then be updated by Equation (4.12).

These steps comprise the basic projection framework for a piping network. The specifics of how the steps are performed depends on how the frictional term \( F \) is modeled. In the present study, two forms of the frictional term are proposed, which give rise to the two methods referred to below as \( P1 \) and \( P2 \).

- **\( P1 \):** \( F \) is modeled only as a function of \( q^n \). This choice results in a first-order explicit forward Euler scheme for which \( F = F(q^n) \). It creates a simple marching method, so that \( q^* \) is given directly by (4.11), \( \phi \) can be explicitly computed from (4.14), and the flow rates and pressures can be updated directly.

- **\( P2 \):** \( F \) is modeled as a function of \( q^{n+1} \). This is a modification of the Van Kan pressure correction for the Navier-Stokes equations [33], and is second-order accurate in the flow rates. The frictional term is linearized by a Newton’s approximation. If \( F(q_{ij}) = \alpha_{ij} q_{ij}^m \), where \( m \geq 1 \), this approximation is \( F = \alpha_{ij} (q_{ij}^n)^{m-1} q_{ij}^{n+1} \). This choice makes the projection method above fully implicit. Now, \( q^* \) cannot be calculated directly from Equation (4.11) as it is dependent on \( q^{n+1} \). Likewise, \( \phi \) is also \( q^{n+1} \) dependent so that when substituted into Equation (4.13), an implicit equation for \( q^{n+1} \) must be solved requiring a matrix inverse at each step. The pressure is then updated as before.
In summary, the algorithm uses the following three steps to advance the solution in time.

**Step 1:** Determine the intermediate flow rate vector $q^*$ using

$$\frac{q^* - q^n}{\Delta t} = -F - M A p^n + M p^n_{i/o}.$$ 

**Step 2:** Determine the pressure correction term $\phi$ necessary to project the flow rate vector onto the manifold of vectors for which $A^T q^{n+1} = 0$ is valid,

$$\phi = \frac{1}{\Delta t} (A^T M A)^{-1} A^T q^* + (A^T M A)^{-1} A^T M (p^{n+1}_{i/o} - p^n_{i/o}).$$ 

**Step 3:** Update the flow rate vector by

$$q^{n+1} = q^* - \Delta t M A \phi + \Delta t M (p^{n+1}_{i/o} - p^n_{i/o})$$ 

followed by the pressure.

4.5 Verification for constant friction factor

In order to verify the accuracy of the two methods numerically, the flow problem in the symmetric self-similar tree network shown in Figure 4.1, using the turbulent with constant friction factor form of $F$ given in (4.7a), is solved. Constant $f$ is chosen because, for a step change in the driving pressure difference, the nonlinear equations have an analytical solution to which the numerical results can be compared. The initial conditions are taken to conform to a steady state at a pressure difference of $\Delta p(0) = 1$ for $t < 0$, and for $t \geq 0$ it changes to $\Delta p(\infty) = 1.5$.

The exact solution of Equation (4.6) with constant friction factor is

$$q_1(t) = \sqrt{\frac{b}{a}} \frac{\Delta p(\infty)}{\Delta p(\infty)} \tanh \left\{ t \sqrt{ab \Delta p(\infty)} + \frac{1}{2} \ln \frac{\sqrt{\Delta p(\infty)} + 1}{\sqrt{\Delta p(\infty)} - 1} \right\}, \quad (4.15)$$
where

\[ a = \begin{cases} \frac{N \left(1 - 2^{-1/2}\right)}{1 - 2^{-N/2}}, & \text{if } \gamma = \sqrt{2}, \\ \frac{4^N - 1}{3N}, & \text{if } \gamma = 2, \\ \frac{2\gamma - 4}{\gamma^4 - 4} \frac{\gamma^{4N} - 4^N}{2^N \gamma^N - 4^N}, & \text{otherwise}. \end{cases} \quad (4.16) \]

\[ b = \begin{cases} \frac{1}{N}, & \text{if } \gamma = 2, \\ \frac{\gamma^{2^{-1}} - 1}{\gamma^N 2^{-N} - 1}, & \text{if } \gamma \neq 2. \end{cases} \quad (4.17) \]

Initial conditions are given by

\[ q_{ij}(0) = \frac{1}{2i-1} \left( \frac{\Delta p(0)}{\sum_{i=1}^{N} \alpha_i \gamma^{i-1} 2^{-n(i-1)}} \right)^{1/2}, \quad (4.18a) \]

\[ \Delta p_{ij} = \gamma^{i-1} 2^{-n(i-1)} \alpha_i q_{11}^n. \quad (4.18b) \]

This problem was also solved numerically using \( P1 \) and \( P2 \) over the time interval \( 0 < t \leq 2 \) which was divided into \( M \) time steps. The error over this interval was calculated using the \( L_2 \) norm using

\[ E = \left\{ \Delta t \sum_n (q_e(t_n) - q^n) \cdot (q_e(t_n) - q^n) \right\}^{1/2}. \quad (4.19) \]

For \( \beta = \sqrt{2} \) and \( N = 5 \), Figure 4.2 shows the error of the flow rate calculations using \( P1 \) and \( P2 \) for increasing \( M \). Method \( P1 \) is seen to be first-order by the log-log plot, as it is clearly seen that an order of magnitude reduction in step size results in an order of magnitude reduction in the error. The same change in
Figure 4.2: Convergence test for $P_1$ (diamonds) and $P_2$ (circles) shows first- and second-order accuracy, respectively.
step size for $P2$ results in a two-order of magnitude in the reduction in the error, indicating second-order accuracy.

While $P2$ offers a modest improvement in accuracy over $P1$, its real benefit is in regards to stability. As with any fully explicit method, $P1$ requires a fairly restrictive time step to retain stability. Depending on both the form of the frictional term and the Newton’s approximation used for linearization, method $P2$ ranges from semi- to fully-implicit and allows for stability with a much larger time step. The disadvantage, though, is that this improved accuracy and stability requires a matrix inversion at each time step. For networks with a very large number of components, this leads to a substantial increase in computational time.

4.6 Flow dynamics

The previous dynamics were computed for a piping network with a quadratic viscous resistance term of the form $F(q_{ij}) \propto q_{ij}^2$ because an analytical solution was available to demonstrate the order of accuracy of each method. The numerical method will now be applied to the tree of Figure 4.1 using the Blasius form of (4.7a) for which there is no analytical solution. Any other friction factor relation can, of course, be used with suitable modifications.

Laminar-turbulent transition affects the dynamic behavior of the flow rate in a network, as will be shown in greater detail below. The simplest situation, however, is if there is no transition in any of the branches, i.e. if the flow in each branch remains either laminar or turbulent.

4.6.1 Without transition

The dynamics of the piping network in Figure 4.1 were computed using the Blasius form of the frictional term for all Reynolds numbers. The dynamics are
then dependent on the branching ratio, $\gamma$, the size of the network, $N$, and of course the pressure difference across the network, $\Delta P$. The effect of varying the network size, $N$, can be seen in Figure 4.3, which shows the dynamics of the inlet flow rate $q = q_{11}$ for networks of increasing size while with a constant branching ratio of $\gamma = \sqrt{2}$ and pressure difference of $\Delta p(\infty) = 1.0^5$ (which corresponds to flows in the range $1000 < Re < 10000$). As would be expected, the change in flow rate for a given change in inlet pressure is much larger for a small network than for a large one; as a network grows in size, the resistance to the flow increases accordingly. This increased head loss across the network causes a smaller gain in flow rate but also allows the network to respond faster and reach a steady-state value quicker. The shorter rise time can be better seen in Figure 4.4 in which the flow rate is normalized by

$$\hat{q} = \frac{q(t) - q(0)}{q(\infty) - q(0)}.$$  \hspace{1cm} (4.20)

The effect of varying $\gamma$ on the normalized flow rate while the size of the network is held constant at $N = 5$, with the same pressure drop as before, is shown in Figure 4.5. In this case an increase in $\gamma$ leads to reduction in the pipe length and diameter. The reduction in length and diameter have opposite effects on the flow resistance, but as the frictional term varies proportionally with the length and inversely proportional to the square of the radius, an increase in $\gamma$ ultimately leads to an increase in flow resistance. As before, the increase in head loss results in shorter rise times but lower flow rates.

Figure 4.6 shows the dynamic response of the inlet flow rate to both a positive and a negative change in the driving pressure. In this case, a step from $\Delta p(0) = 4.0 \times 10^4$ to $\Delta p(\infty) = 1.0^5$ is the positive step, and from $\Delta p(0) = 1.0^5$ to $\Delta p(\infty) = 4.0^4$ is the negative. Figure 4.7 shows the normalized flow rate. Interestingly, the responses are not identical; the nonlinearity of the friction term leads to slightly
Figure 4.3: Time-dependent flow rate due to step change in inlet pressure for networks of varying size.
Figure 4.4: Normalized time-dependent flow rate due to step change in inlet pressure for networks of varying size.
Figure 4.5: Time-dependent flow rate due to step change in inlet pressure for networks of varying $\gamma$. 

\[ \dot{q} \]

Increasing $\gamma$.
shorter rise than fall times.

4.6.2 With transition

In real world pipe flows, the frictional term is not a continuous function of the flow rate, but instead depends on whether the flow is laminar or turbulent. This results in a piecewise formula with a transition region. The network flow problem then becomes much more rich dynamically as each pipe has the potential to transition from one regime to the other which results in a change in flow resistance, which in turn affects the flow through each of the other pipes in the network.

To examine the effect of transition, the tree network was analyzed using the laminar and Blasius turbulent friction factors in Equation (4.7a), where the critical
Figure 4.7: Normalized time-dependent flow rate due to positive (solid line) and negative (dotted line) step changes in inlet pressure.
Reynolds number for transition was taken to be \( Re_c = 3500 \) and hysteresis effects were ignored. Since the Reynolds number in each branch depends on \( \gamma \) as \( Re_{ij} = \left(\frac{4q_{11}}{\pi}\right)\left(\frac{\gamma}{2}\right)^{-1} \), three possibilities exist for the network. For \( \gamma < 2 \), \( Re_{ij} \) increases with generation number \( i \). For \( \gamma = 2 \), \( Re_{ij} \) is independent of generation number and remains the same for all pipes in the network. For \( \gamma > 2 \), \( Re_{ij} \) decreases with generation number \( i \). In each of these three possibilities, transition will first occur in different places as the flow rate increases. For \( \gamma < 2 \), transition first occurs in the inlet pipe, while for \( \gamma > 2 \), transition will first occur in the last generation of pipes. For \( \gamma = 2 \), all pipes will transition simultaneously. For a network operating at a steady state with laminar flow through all branches, a positive step-change in inlet pressure will result in an increase in flow rate through all pipes. A large enough step-change can cause transition to cascade from one side of the network to the other.

The overall pressure difference \( \Delta p \) across the network will then determine the network’s final state. Not all \( \Delta p \)’s will result in a steady final state. For certain \( \Delta p \)’s, the flow in some pipes, or even in the entire network, may oscillate between the laminar and turbulent regimes. The oscillations are due to the fact that when \( Re_c \) is reached, the transition from laminar to turbulent flow causes a large increase in the viscous resistance term of Equation (4.4). If the increase in the resistance term is larger than the pressure force, the fluid will decelerate until it reaches \( Re_c \) and then transition back to laminar flow, and a pattern of laminar-turbulent oscillations is established. This is not a network-related phenomenon, but can be observed even for flow in a single pipe using a simple capillary tube experiment where an overhead plenum produces a pressure driven flow. However, when these transition oscillations exist in even a single pipe within a network, oscillations in flow rate will occur across every other branch through the coupling of the flow.
rates and nodal pressures. This is a network-related phenomenon, as flow rate vac-
illations without transition will not occur in a single pipe with a constant pressure
drop; neither steady state analysis nor dynamic analysis of individual components
would predict it.

Tables 4.1, 4.2, and 4.3, show the possible final states for an $N = 5$ network
with $\gamma = 1.5$, 2, and 2.5, respectively, for different $\Delta p$. The states were observed
by applying $P2$ for different $\Delta p$ and the ranges were verified analytically. In Table
4.1 it is seen that as $\Delta p$ increases, each generation, from inlet to outlet, in turn
transitions from laminar to oscillatory to turbulent giving 11 possible long term
solutions. Table 4.3 show similar states, but proceeding from outlet to inlet as $\Delta p$
increases. Table 4.2 shows that for $\gamma = 2.5$ only three solutions are possible: all
flows are laminar, oscillatory, or turbulent. The most interesting cases are those in
which oscillations occur or when the transition from laminar to turbulent cascades
completely across the network.

An example of laminar/turbulent oscillations, corresponding to Case 2 of Table
4.1, is shown in Figure 4.8. With an overall pressure difference between inlet and
outlet of $\Delta p = 5.0 \times 10^5$, the flow in the inlet pipe nearly reaches a steady-state
as the head loss due to the laminar friction approaches the absolute pressure drop
across it. But when the the flow transitions to turbulent, the flow resistance
increases sharply such that the head loss is now greater than the pressure drop
driving the flow and the fluid decelerates until again reaching $Re_c$ and becoming
laminar again. The process then repeats.

The turbulent cascade is shown in Figure 4.9, corresponding to Case 11 in Table
4.1. In this example, each generation consecutively transitions from laminar to
turbulent flow, beginning with the inlet pipe in the first generation and continuing
to the outlet pipes in the fifth generation. The time between transitions depends
Figure 4.8: Non-dimensional inlet flow rate with $\Delta P = 5.0 \times 10^5$. Flow oscillates between laminar and turbulent regimes.
on $\Delta p(\infty)$, but increases between each successive generation.

### 4.7 Conclusions: numerical analysis

With a numerical method such as the one proposed here, the dynamics of a piping network can be studied as system parameters and driving pressure differences are varied. The network can then be optimized to meet design criteria. Two numerical algorithms were proposed, both extensions of Chorin’s projection method developed for the solution of the nonlinear DAE system resulting from the Navier-Stokes equations. The first method, $P1$, is first-order accurate as uses a
Figure 4.9: Non-dimensional flow rates for each generation with $N = 5$, $\gamma = 1.5$, $\Delta p = 5.0 \times 10^5$. Vertical dotted lines signify instant at which transition occurs. $\triangle$, $\star$, $\diamond$, $\ast$ and $\bullet$ denote flows in generations 1, 2, 3, 4 and 5, respectively.
### Table 4.2

**Possible Solutions for \( \gamma = 2 \)**

L = Laminar, O = Oscillating, T = Turbulent

<table>
<thead>
<tr>
<th>Case</th>
<th>( \Delta p )</th>
<th>Generation number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 0 &lt; \Delta p \leq 9.3737 \times 10^5 )</td>
<td>L L L L L</td>
</tr>
<tr>
<td>2</td>
<td>( 9.3737 \times 10^5 &lt; \Delta p \leq 2.1088 \times 10^6 )</td>
<td>O O O O O</td>
</tr>
<tr>
<td>3</td>
<td>( 2.1088 \times 10^6 &lt; \Delta p )</td>
<td>T T T T T</td>
</tr>
</tbody>
</table>

The forward-Euler approach. The second, \( P2 \), also employs the forward-Euler time discretization, but uses a Newton’s approximation for the nonlinear term to achieve second-order accuracy in the flow rates and increased stability. The accuracy of both was confirmed by computing the response to a step change in pressure difference of a self-similar tree-like piping network with a quadratic flow resistance term and comparing to the available analytical solution.

The dynamics of a self-similar tree-like piping network were also computed using the the Blasius friction factor for the turbulent regime. The effects of varying system parameters \( \gamma \), \( N \), and \( \Delta p \) were shown. The effects of transition were observed, including cascading transition and laminar/turbulent oscillations. The range of final states for the three possible cases of the geometrical scaling ratio \( \gamma = 2 \), \( \gamma > 2 \), and \( \gamma < 2 \) were tabulated. Emerging behavior that is not observable in steady-state network analysis or predictable from the dynamic analysis of individual components, such as the appearance of oscillating flow rates in pipes.
TABLE 4.3

POSSIBLE SOLUTIONS FOR $\gamma = 2.5$

$L = \text{LAMINAR}, \ O = \text{OSCILLATING}, \ T = \text{TURBULENT}$

<table>
<thead>
<tr>
<th>Case</th>
<th>$\Delta p$</th>
<th>Generation number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$i = 1$</td>
</tr>
<tr>
<td>1</td>
<td>$0 &lt; \Delta p \leq 4.8100 \times 10^6$</td>
<td>L</td>
</tr>
<tr>
<td>2</td>
<td>$4.8100 \times 10^6 &lt; \Delta p \leq 1.0051 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>3</td>
<td>$1.0051 \times 10^7 &lt; \Delta p \leq 1.4713 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>4</td>
<td>$1.4713 \times 10^7 &lt; \Delta p \leq 1.5552 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>5</td>
<td>$1.5552 \times 10^7 &lt; \Delta p \leq 2.2959 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>6</td>
<td>$2.2959 \times 10^7 &lt; \Delta p \leq 2.3094 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>7</td>
<td>$2.3094 \times 10^7 &lt; \Delta p \leq 3.4123 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>8</td>
<td>$3.4123 \times 10^7 &lt; \Delta p \leq 3.4144 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>9</td>
<td>$3.4144 \times 10^7 &lt; \Delta p \leq 5.0455 \times 10^7$</td>
<td>L</td>
</tr>
<tr>
<td>10</td>
<td>$5.0455 \times 10^7 &lt; \Delta p \leq 5.0459 \times 10^7$</td>
<td>O</td>
</tr>
<tr>
<td>11</td>
<td>$5.0459 \times 10^7 &lt; \Delta p$</td>
<td>T</td>
</tr>
</tbody>
</table>

without laminar-to-turbulent transition, demonstrates the importance of dynamic analysis in flow networks.

While a self-similar branching tree was used here, the method is applicable to piping networks of any configuration. Different configurations will result in alternate incidence matrices, although still formed by the same rules. It should also be noted that the data reported is specific to the geometry and friction factor formulas being used, in this case the Darcy and Blasius friction formulas, but there is no restriction on the friction formulas that can be used. Additional piping network components such as bends or valves can be added to supplement the model.
in order to study more complex networks. These components can be modeled using equations 4.6a and 4.6b and added directly into the matrix equations 4.9a and 4.9b. The solution procedure remains unchanged. Dynamic effects, such as hysteresis during transition, can be incorporated as well by using more advanced friction factor formulas.
CHAPTER 5

APPROXIMATION OF GRID-LIKE NETWORKS

Previous discussion has been focused on, but not limited to, a specific class of self-similar systems. In Chapter 3, self-similar bifurcating trees were considered in detail and a solution technique was presented. More generally the analysis of Chapter 3 is applicable to $n$-furcating trees, but as a matter of convenience and brevity, discussion was for the most part focused on bifurcating systems. It should be remembered that over the development of that chapter, an assumption of linearity was made. In the previous chapter, the linear assumption was lifted and non-linear bifurcating networks were considered in detail. While discussion remained focused on bifurcating trees, again, the analysis remains equally valid for networks of other geometries as well. In this chapter, another class of self-similar network will be considered and a few possible methods of reduction will be considered.

5.1 Grid-like networks

While tree-like networks appear with great frequency in the natural world, there exist and endless number of alternate network geometries that could be studied. As such there exists a wealth of literature available on network topology, formulation, and behavior [54, 62]. Even if one restricted oneself to networks exhibiting
self-similarity, there still exist enough unique geometries and structures to fill an academic career. Researchers such as Barabasi [5] and Bejan [6] have done just that. But for the purposes of this chapter, we will again choose a specific class of networks to consider - grid-like networks.

Grid-like networks, like the bifurcating trees of earlier chapters, are a commonly occurring self-similar network. In many ways, this is a topology even more prevalent in engineering than are bifurcating networks. They can be seen on any map as a network of alleys, streets, highways and interstates. They are commonly used for HVAC systems, city-wide plumbing and piping networks, and electrical distribution networks. And yet, despite the frequency in which they appear within engineering systems, they can still exhibit unexpected behavior. For example, disturbances within the U.S. power grid can sometimes lead to unexpected cascading failures [17] as occurred in parts of 14 western states in July 1996 [14] and across New York and much of the northeast in 2003 [27].

Another reason for choosing a self-similar grid-like network for study is the key property that distinguishes it from the tree-like networks of previous chapters. In a bifurcating tree network, flow is always in one direction - from inlet to outlet. In a grid-like network, there exists the possibility for closed loop flows meaning that there are closed paths within the network itself. In a tree network, once flow has exited from a branch there is no path that can be chosen through the network to return back to the branch. In this sense, a grid-like network can be considered in some ways to be the two-dimensional extension of a tree network.

Figure 5.1 is a schematic of a self-similar network. As it is necessary to make some conditions on the networks topology, the grid-like networks considered here are assumed to be a square network of pipes with inlets at the four corners and outlets at regular points within its interior. Figure 5.2 is an example of a first
generation grid-like network with four inlets and one outlet. This network has a characteristic length scale, $L$, so each connecting pipe is of length $L/2$. Also, it is assumed that there is similarity in the diameter of the pipes between generations. The diameter of all of the pipes for the first generation network is assumed $D$.

The second generation is given by creating a network similar to the first generation within each internal square of the first generation network. Each pipe in the second generation is now of length $L/4$, but the diameters of the pipes within the network can now vary. Each of the original pipes (those present in generation one) are diameter $D$, but the new pipes added to create the second generation are diameter $D/\beta$. The second generation network is shown in Figure 5.3. The $n^{th}$ generation is given by recursively applying the above. The diameter of the pipes added to create the $n^{th}$ generation is then $D/\beta^{k-1}$, where $k$ is the generation where the pipe was added, and the length of every pipe in the network is $L/2^n$. This is the topology of the networks being considered throughout this chapter.

Assuming that the pressures at the four inlets and each of the outlets are known,
Figure 5.2: Schematic of a first-generation network.

Figure 5.3: Schematic of a second-generation network.
then to acquire a full dynamic solution to the network requires one to solve for all of the resulting flow rates in, out, and within the network as well as the pressure at every node. To do this requires the same mathematical modeling of previous chapters; a continuity equation can be written for every node and a momentum equation can be written for every branch of pipe connecting any two nodes. The solution to the resulting DAE system fully describes the dynamics of the network.

For the first generation network the above procedure yields nine algebraic continuity equations and twelve differential equations conserving momentum, resulting in coupled system of twenty-one equations. The unknowns then become the five inlet and outlet flow rates, the twelve branch flow rates, and the four pressures at the nodes that are not inlets or outlets. This gives twenty-one equations with twenty-one unknowns; a well posed problem that can be directly solved.

For the second generation network, the same four inlet pressures are known and sixty-five equations and unknowns result. This time there are forty momentum equations, twenty-five continuity equations, eight unknown inlet/outlet flow rates, seventeen unknown pressures, and forty unknown flow rates.

Extending this to a full $N$-generation network results in a large system of differential algebraic equations: $(2^n + 1)^2$ algebraic continuity equations (one for each node) and $(2^{n+1})(2^n + 1)$ differential momentum equations (one for each branch). As $N$ increases, the size of the DAE system to be solved increases exponentially, making it computationally expensive to obtain a dynamic solution even after just a few generations. Table 5.1 shows how the size of the network and the resulting DAE system grow with increasing generation number. Because of the exponential growth of the network, any additional methods of reduction are useful to simplify analysis.
TABLE 5.1

THE NUMBER OF INLETS, OUTLETS, BRANCHES, AND NODES FOR NETWORKS OF INCREASING SIZE

<table>
<thead>
<tr>
<th>Generations</th>
<th>Inlets</th>
<th>Outlets</th>
<th>Branches/ Nodes/ Generations</th>
<th>Differential Equations</th>
<th>Algebraic Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>12</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>40</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>16</td>
<td>144</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>64</td>
<td>544</td>
<td>289</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>4</td>
<td>$2^{(n-1)}$</td>
<td>$(2^{n+1})(2^n + 1)$</td>
<td>$(2^n + 1)^2$</td>
<td></td>
</tr>
</tbody>
</table>

Governing equations

Following Franco [21] and the analysis of previous sections, using a one dimensional flow model and assuming fully developed flow while neglecting inlet and exit effects, the momentum equations for a branch is given as

$$\frac{dq}{dt} + F(q) - \frac{\pi D_k^2}{4 \rho L_n} \Delta p = 0,$$

where $q$ is the flow rate through the branch, $\Delta p$ is the pressure drop across the branch, $\rho$ is fluid density and $t$ is time. The resistance term is different for laminar and turbulent flow and is given by as

$$F(q) = \begin{cases} 
32 \nu q / D_k^2 & \text{for } Re < Re_c \text{ (laminar flow)} \\
0.1894 \nu^{1/4} q^{7/4} / D_k^{11/4} & \text{for } Re \geq Re_c \text{ (turbulent flow)}
\end{cases}$$

The critical Reynolds number ($Re_c$) at which the flow transitions from laminar to turbulent is given as $Re = 4q / \pi D_k \nu$. The continuity equation for each node is
given by
\[ \sum q_{\text{in}} - \sum q_{\text{out}} = 0. \] (5.3)

Nondimensionalizing Equation (5.1) and (5.3) by viscous pressure, time, and flow scales, the momentum equation becomes
\[ \frac{dq}{dt} + F(q) - 2^n \beta^{-2(k-1)} \Delta p = 0, \] (5.4)

where
\[ F(q) = \begin{cases} 
\beta^{2(k-1)}q & \text{for } Re < Re_c \text{ (laminar flow)} \\
\alpha \beta^{11/4(k-1)} q^{7/4} & \text{for } Re \geq Re_c \text{ (turbulent flow)}
\end{cases} \] (5.5)

The non-dimensional continuity equation is now
\[ \sum q_{\text{in}} - \sum q_{\text{out}} = 0. \] (5.6)

When considering very large or infinite networks, it is sometimes necessary to ensure that all quantities actually remain finite. In this case, the total length of pipe within an \( n \)th generation network is given by
\[ L(n) = L(2^{n+1} + 2), \] (5.7)

thus the length of pipe present in a network grows exponentially with generation number. (Figure 5.4). The total volume contained within the pipe network can also be calculated and is given by
\[ V(n) = L \pi D_0^2 \left( 6 + \sum_{k=2}^{k=n} \frac{2^k}{\beta^{2(k-1)}} \right). \] (5.8)

Figure 5.5 shows that for certain values of \( \beta \) the total network volume is finite. For the volume to be finite as \( n \to \infty \) the cross sectional area of the pipes must go to zero faster than the length is going to infinity, that is
\[ \lim_{n \to \infty} \frac{2^n}{\beta^{2(n-1)}} = 0. \] (5.9)

For this to be true, \( \beta > \sqrt{2} \).
Figure 5.4: Total network length vs. generation number.
Figure 5.5: Network volume vs. generation number.
5.2 Reduction techniques

Several methods will be considered in the following sections for reducing the self-similar grid-like networks described above. Each of these methods can be used alone or in combination to simplify analysis and are intended to serve as a short survey of possible methods of reduction than can be used not only for grid-like networks, but for the more general class of networks exhibiting self-similarity. As such, none of the methods are developed in great detail. The first method that will be considered takes advantage of the principle of linear superposition. This will allow much simpler solutions of a self-similar grid-like network with slightly asymmetric boundary conditions. Secondly, we will briefly revisit a method considered in Chapter 3: reduction by studying the behavior of a network as it grows in size to approximate behavior of a much larger network. Finally, we will consider the lines of symmetry that exist within a grid-like network and how they can be used to simplify analysis.

For all of the analysis performed, the following notation and assumptions will be used:

- The differential operators governing transfer through each of the branches is assumed linear - only the laminar flow case of Equation 5.5 will be considered.
- The DAE equation sets can be written in the frequency or Laplace domain such that \( \mathcal{L} q \) represents the Laplace transform of \( L q \), where \( L \) is the linear operator governing the flow, i.e., for one particular case,

\[
L q = \frac{dq}{dt} + q,
\]

and

\[
\mathcal{L} q = s + 1.
\]

- Numbering of branches is done from left-to-right, top-to-bottom. Nodes are identified by the connected branch with the lowest indices. Numbering is illustrated in Figure 5.15.

- For consistency, all flows are considered to be from left-to-right and top-to-bottom. In terms of indices, the direction of flow in a branch is given as into the higher indices node and out of the lower.
5.2.1 Using linearity and the principle of superposition

If the operators governing the transport through each branch are linear (such as is the case for laminar flow), superposition can be used to aid in reduction. A network with four different inlet pressures (inhomogeneous conditions) can be decomposed into four individual networks as shown in Figure 5.6, each with only one inlet. The net response or solution at any given point or time can then be given by combining the solutions of the four individual networks. In this way a simplified network can be solved with four different boundary conditions and their solutions can be superimposed to produce the solution to the original, inhomogeneous network.

This is a method commonly used for solving 2-D heat conduction problems with multiple inhomogeneous boundary conditions or inhomogeneous differential equations as occurs with internal heat generation. In such examples, the standard solution method of separation of variables is not applicable as the ODE and boundary conditions do not fit the Sturm-Liouville conditions of a homogeneous differential equation with three homogeneous and one inhomogeneous boundary conditions.

Inlets are marked with an open circle, outlets are marked with a closed circle. Outlets are numbered left-to-right, top-to-bottom, starting from $q_{out,1}$.
condition. The principle of superposition, however, states that a linear system is additive - that is, any sum of solutions to the differential equation is also a solution to the original differential equation. In this way the system, and its multiple inhomogeneous boundary conditions, can be broken into several Sturm-Louisville problems that when summed together, satisfy the original boundary conditions and differential equation.

Oftentimes breaking the system into a series of subsystems which sum to the original system can take some creativity. In the case of the grid-line network being considered, the subnetworks are somewhat more obvious.

5.2.2 Using self-similarity

Another approach to simplification is to take advantage of the similarity between one generation and the next. If the solution for an $N$-generation network can be related to the solution of an $(N - 1)$-generation network, then the solution of a network of arbitrary generation can be determined by starting with a known solution and successively approximate larger and larger networks. Similarly, if one can study a network and determine that as the network grows in size the solution tends to converge to quickly converge to a ‘large network’ solution, very large networks can be approximated by much smaller and simpler models. To demonstrate this, a few simple flow quantities of the self-similar grid-like networks described in Section 5.1 will be considered.

As was previously noted, for laminar flow the resulting equations are linear. For turbulent flow, the equations are non-linear because of the form of viscous resistance term $F(q)$. For either case, considering unsteady flow results in a system of differential algebraic equations (DAE’s). For simplicity, consider the steady, laminar flow case in which the governing equations can be written as $Ax = b$ and

106
can be solved by matrix inversion. For large systems ($n > 5$), numerical techniques for matrix inversion can be used, such as Jacobian or Gauss-Seidel methods.

For the steady laminar case with a given pressure drop between the inlets and outlets (sinks), the total flow rate through the network can be calculated for different values of $n$ (generation number) and $\beta$. Figure 5.7 shows that for certain values of $\beta < \beta_c$ the flow rate increases with $n$ while for other $\beta > \beta_c$ the total flow rate decreases with $n$. Similarly, the total network resistance, which can be thought of as the ratio between the pressure drop and the induced flow rate, or $r = \Delta P/q$, increases for some values of $\beta$ and decreases for others (Figure 5.8).
Figure 5.8: Network resistance vs. generation number.
There are also values of $\beta$ that cause the flow rate to increase for some increases in generation number $n$ and decrease for other changes in generation number. These values of $\beta$ create critical $\beta$-$n$ curves that define the behavior of the system. One interesting curve in the $\beta$-$n$ plane is the set of values where the total flow rate does not change with a unit increase in network generation. Similarly, there is a different curve in the $\beta$-$n$ plane where the total flow rate through the network remains unchanged from the first generation to generation $n$. Figure 5.9 shows the two curves.

Both of these lines are about $\beta \approx 1.6$. This seems to a rough estimate of the
value of $\beta$ that defines the behavior of the system. For $\beta > 1.6$, the total flow rate decreases with increasing $n$, while for $\beta < 1.6$, the total flow rate tends to increase with the generation number.

In addition to the effect of $\beta$ and $n$, the distribution of flow can also be considered. The distribution of the outlet flow rates (through the sinks) also varies with $\beta$ and $n$. For small $\beta$ values ($\beta < 1.3$), the flow rates at the sinks nearest to the four inlets account for the majority of the total outlet flow (Figure 5.11). For larger values, $\beta > 1.6$, the total outlet flow is more evenly distributed between all of the sinks. Figure 5.10 shows the effect $\beta$ has on the distribution of the outlet flow.
flows. As $\beta$ increases, the local outlet flows become much more uniform. Figures 5.12 and 5.13 show the standard deviation and the mean, respectively, of all of the local outlet flows as a function of $n$ for several different values of $\beta$.

The effect of $\beta$ on the flow rate is the opposite of what it is on the flow distribution. If the desire is uniform outlet flow, then the overall flow rate will be sacrificed. This suggests an optimum $\beta$ for a maximum flow rate and uniform distribution that depends on the relative weight placed on each goal.

However, for all values of $\beta$, it can be observed that regardless of network size, the overall flow distribution follows a general pattern. For example in Figure 5.10, the distribution between generation $N = 3$ and $N = 4$ for a given value of $\beta$ varies
Figure 5.12: Standard deviation of the local outlet flow rates
Figure 5.13: Mean value of local flow rates as a function of $n$
little. This suggests that as a network grows, one could predict the outlet flow by first estimating the total flow and then applying the flow distribution given by the system’s $\beta$ value. In this case, it is not necessary to compute the full solution for a very large network as flow distribution and total flow values can be approximated by first studying the behavior of much smaller networks.

5.2.3 Using symmetry

Another approach that can be used to simplify network analysis is to take advantage of symmetry within the system. Symmetry is a natural property of self-similar systems and for the grid-like networks being considered, axes of symmetry are readily apparent. As a quick note, for this section we will redefine our concept of a first-generation grid-like network. We will offset our generation index by one so that what was previously defined as a second generation network, we will now call an $N = 1$ generation network.

For a network with four identical inlet pressures, there are 4 lines of similarity. For a network with only one inlet, there is one. For either of these cases, the entire network, and thus the entire DAE system, is not required to obtain a solution. Instead, only the smallest segment between lines of similarity is needed. Figure 5.14 shows the lines and the smallest segments needed to obtain a solution. In the following subsections, a one-, two-, and three- generation network are analyzed. It is assumed that the pressure at the four inlets are identical as well as those at the outlets. With these assumptions, only one-eighth of each of the networks (and the DAE system) are needed to obtain a complete solution.
Figure 5.14: The lines of similarity in the network. The shaded area represents the portion of the network required for a solution.

First Generation Network

A 1-generation network and its 1/8th network model is shown in Figure 5.16. Symmetric boundary conditions (for the flow rates) are used across each line of symmetry resulting in system of equations shown in Table 5.2.

The goal is to solve for the outlet flow, $q_{out}$, as a function of the overall pressure drop $\Delta P_{i,o} = P_{in} - P_{out}$. For the system of equations generated from a first generation network, the outlet flow can be solved for in Mathematica. In the Laplace domain, it can be written as

$$q_{out} = \frac{10}{9L} (P_{in} - P_{out}).$$

Second Generation Network

A 2-generation network and its 1/8th network model is shown in Figure 5.17. Again, symmetric boundary conditions are used and can be seen in Figure 5.17(b).
As can be seen, there are now 3 outlet flows in the $1/8^{th}$ model, which we will denote $q_{out,1}$, $q_{out,2}$, and $q_{out,3}$. Outlets are numbered in the same manner as branches, left-to-right and top-to-bottom. The resulting system of equations is shown in Table 5.3.

Again, the goal is to derive the outlet flows $(q_{out,1}, q_{out,2}, q_{out,3})$ as a function
TABLE 5.2

THE DAE SYSTEM DESCRIBING THE 1/8TH MODEL OF A FIRST GENERATION NETWORK

<table>
<thead>
<tr>
<th>Differential</th>
<th>Algebraic</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L q_1 = P_{in} - P_1 )</td>
<td>( q_1 = q_2 + q_3 )</td>
</tr>
<tr>
<td>( L q_2 = P_1 - P_2 )</td>
<td>( 2q_2 = q_4 )</td>
</tr>
<tr>
<td>( L q_3 = P_1 - P_{out} )</td>
<td>( 2q_3 = 2q_5 + q_{out} )</td>
</tr>
<tr>
<td>( L q_4 = P_2 - P_4 )</td>
<td>( 2q_5 + q_4 = q_6 )</td>
</tr>
<tr>
<td>( L q_5 = P_{out} - P_4 )</td>
<td>( 4q_6 = 0 )</td>
</tr>
<tr>
<td>( L q_6 = P_4 - P_6 )</td>
<td></td>
</tr>
</tbody>
</table>

of the pressure drop \( \Delta P_{i,o} = P_{in} - P_{out} \). For the second generation network, the solution can be written in the Laplace domain as

\[
q_{out,1} = \frac{2151}{2435L} (P_{in} - P_{out}),
\]

\[
q_{out,2} = \frac{2294}{14610L} (P_{in} - P_{out}),
\]

\[
q_{out,3} = \frac{197}{7305L} (P_{in} - P_{out}).
\]

Third Generation Network

A 3-generation network and its 1/8th equivalent network model are shown in Figures 5.18 and 5.19, respectively. The number of outlets (marked by solid circles in Figure 5.19) jumps from 3 to 10 from the second to the third generation. Applying symmetric boundary conditions as before, the resulting set of DAEs is shown in Table 5.4. This system can be solved for the outlet flows \( (q_{out,1}, q_{out,2}, \ldots, q_{out,10}) \) as a function of the overall pressure difference \( P_{in} - P_{out} \) as before. The results can
TABLE 5.3

THE DAE SYSTEM DESCRIBING THE 1/8TH MODEL OF A SECOND GENERATION NETWORK

<table>
<thead>
<tr>
<th>Differential</th>
<th>Algebraic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L q_1 = P_{in} - P_1$</td>
<td>$q_1 = q_2 + q_3$</td>
</tr>
<tr>
<td>$L q_2 = P_1 - P_2$</td>
<td>$q_2 = q_3 + q_6$</td>
</tr>
<tr>
<td>$L q_3 = P_2 - P_3$</td>
<td>$q_3 = q_4 + q_7$</td>
</tr>
<tr>
<td>$L q_4 = P_3 - P_4$</td>
<td>$2q_4 = q_8$</td>
</tr>
<tr>
<td>$L q_5 = P_1 - P_{out,1}$</td>
<td>$2q_5 = 2q_9 + q_{out,1}$</td>
</tr>
<tr>
<td>$L q_6 = P_2 - P_6$</td>
<td>$q_6 + q_9 = q_{10} + q_{12}$</td>
</tr>
<tr>
<td>$L q_7 = P_3 - P_{out,2}$</td>
<td>$q_7 + q_{10} = q_{11} + q_{13} + q_{out,2}$</td>
</tr>
<tr>
<td>$L q_8 = P_4 - P_8$</td>
<td>$q_8 + 2q_{11} = q_{14}$</td>
</tr>
<tr>
<td>$L q_9 = P_{out,1} - P_6$</td>
<td>$2q_{12} = 2q_{15}$</td>
</tr>
<tr>
<td>$L q_{10} = P_6 - P_{out,2}$</td>
<td>$q_{13} + q_{15} = q_{16} + q_{17}$</td>
</tr>
<tr>
<td>$L q_{11} = P_{out,2} - P_8$</td>
<td>$q_{14} + 2q_{16} = q_{18}$</td>
</tr>
<tr>
<td>$L q_{12} = P_6 - P_{12}$</td>
<td>$2q_{17} = 2q_{19} + q_{out,3}$</td>
</tr>
<tr>
<td>$L q_{13} = P_{out,2} - P_{13}$</td>
<td>$q_{18} + 2q_{19} = q_{20}$</td>
</tr>
<tr>
<td>$L q_{14} = P_8 - P_8$</td>
<td>$4q_{20} = 0$</td>
</tr>
<tr>
<td>$L q_{15} = P_{12} - P_{13}$</td>
<td></td>
</tr>
<tr>
<td>$L q_{16} = P_{13} - P_{14}$</td>
<td></td>
</tr>
<tr>
<td>$L q_{17} = P_{13} - P_{out,3}$</td>
<td></td>
</tr>
<tr>
<td>$L q_{18} = P_{14} - P_{18}$</td>
<td></td>
</tr>
<tr>
<td>$L q_{19} = P_{out,3} - P_{18}$</td>
<td></td>
</tr>
<tr>
<td>$L q_{20} = P_{18} - P_{20}$</td>
<td></td>
</tr>
</tbody>
</table>
be expressed in the form

$$q_{out,N} = \frac{C_N}{L} (P_{in} - P_{out})$$

(5.14)

and are shown in Table 5.5. It should be remembered that the numbering of outlets is done first right-to-left and then top-to-bottom such that $C_3$ for a two-generation network is not physically located in the same position as $C_3$ in a three- or four-generation network. However, what is notable about the outlet flows shown in Table 5.5 is the small number of generations needed for outlet flows, and the total outlet flow, to converge.

5.3 Conclusions: reduction of other self-similar networks

The three previous methods of reduction proposed are by no means the only possible methods at the disposal of one interested in modeling grid-like networks,
TABLE 5.4
THE DAE SYSTEM DESCRIBING THE 1/8\textsuperscript{TH} MODEL OF A THIRD GENERATION NETWORK

<table>
<thead>
<tr>
<th>Differential</th>
<th>Differential</th>
<th>Algebraic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L q_1 = P_{16} - P_1$</td>
<td>$L q_{37} = P_{25} - P_{out,5}$</td>
<td>$q_1 = q_2 + q_9$</td>
</tr>
<tr>
<td>$L q_2 = P_3 - P_2$</td>
<td>$L q_{38} = P_{26} - P_{38}$</td>
<td>$q_2 = q_3 + q_{10}$</td>
</tr>
<tr>
<td>$L q_3 = P_2 - P_3$</td>
<td>$L q_{39} = P_{27} - P_{out,6}$</td>
<td>$q_3 = q_4 + q_{11}$</td>
</tr>
<tr>
<td>$L q_4 = P_3 - P_4$</td>
<td>$L q_{40} = P_{28} - P_{40}$</td>
<td>$q_4 = q_5 + q_{12}$</td>
</tr>
<tr>
<td>$L q_5 = P_4 - P_5$</td>
<td>$L q_{41} = P_{29} - P_{out,7}$</td>
<td>$q_5 = q_6 + q_{13}$</td>
</tr>
<tr>
<td>$L q_6 = P_5 - P_6$</td>
<td>$L q_{42} = P_{30} - P_{42}$</td>
<td>$q_6 = q_7 + q_{14}$</td>
</tr>
<tr>
<td>$L q_7 = P_6 - P_7$</td>
<td>$L q_{43} = P_{out,5} - P_{38}$</td>
<td>$q_7 = q_8 + q_{15}$</td>
</tr>
<tr>
<td>$L q_8 = P_7 - P_8$</td>
<td>$L q_{44} = P_{38} - P_{out,6}$</td>
<td>$2q_8 = q_{16}$</td>
</tr>
<tr>
<td>$L q_9 = P_8 - P_{out,1}$</td>
<td>$L q_{45} = P_{out,6} - P_{40}$</td>
<td>$2q_9 = 2q_{17} + q_{out,1}$</td>
</tr>
<tr>
<td>$L q_{10} = P_3 - P_{out,10}$</td>
<td>$L q_{46} = P_{40} - P_{out,7}$</td>
<td>$q_{10} + q_{17} = q_{18} + q_{24}$</td>
</tr>
<tr>
<td>$L q_{11} = P_5 - P_{out,2}$</td>
<td>$L q_{47} = P_{out,7} - P_{42}$</td>
<td>$q_{11} + q_{18} = q_{19} + q_{25} + q_{out,2}$</td>
</tr>
<tr>
<td>$L q_{12} = P_3 - P_{12}$</td>
<td>$L q_{48} = P_{38} - P_{48}$</td>
<td>$q_{12} + q_{19} = q_{20} + q_{26}$</td>
</tr>
<tr>
<td>$L q_{13} = P_5 - P_{out,3}$</td>
<td>$L q_{49} = P_{out,6} - P_{49}$</td>
<td>$q_{13} + q_{20} = q_{21} + q_{27} + q_{out,3}$</td>
</tr>
<tr>
<td>$L q_{14} = P_6 - P_{14}$</td>
<td>$L q_{50} = P_{40} - P_{50}$</td>
<td>$q_{14} + q_{21} = q_{22} + q_{28}$</td>
</tr>
<tr>
<td>$L q_{15} = P_7 - P_{out,4}$</td>
<td>$L q_{51} = P_{out,7} - P_{51}$</td>
<td>$q_{15} + q_{22} = q_{23} + q_{29} + q_{out,4}$</td>
</tr>
<tr>
<td>$L q_{16} = P_8 - P_{16}$</td>
<td>$L q_{52} = P_{42} - P_{52}$</td>
<td>$q_{16} + 2q_{23} = q_{30}$</td>
</tr>
<tr>
<td>$L q_{17} = P_{out,1} - P_{10}$</td>
<td>$L q_{53} = P_{48} - P_{49}$</td>
<td>$2q_{24} = 2q_{31}$</td>
</tr>
<tr>
<td>$L q_{18} = P_{10} - P_{out,2}$</td>
<td>$L q_{54} = P_{49} - P_{50}$</td>
<td>$q_{25} + q_{31} = q_{32} + q_{37}$</td>
</tr>
<tr>
<td>$L q_{19} = P_{out,2} - P_{12}$</td>
<td>$L q_{55} = P_{50} - P_{51}$</td>
<td>$q_{26} + q_{32} = q_{33} + q_{38}$</td>
</tr>
<tr>
<td>$L q_{20} = P_{12} - P_{out,3}$</td>
<td>$L q_{56} = P_{51} - P_{52}$</td>
<td>$q_{27} + q_{33} = q_{34} + q_{39}$</td>
</tr>
<tr>
<td>$L q_{21} = P_{out,3} - P_{14}$</td>
<td>$L q_{57} = P_{40} - P_{out,8}$</td>
<td>$q_{28} + q_{34} = q_{35} + q_{40}$</td>
</tr>
<tr>
<td>$L q_{22} = P_{out,4} - P_{out,4}$</td>
<td>$L q_{58} = P_{50} - P_{58}$</td>
<td>$q_{29} + q_{35} = q_{36} + q_{41}$</td>
</tr>
<tr>
<td>$L q_{23} = P_{out,4} - P_{24}$</td>
<td>$L q_{59} = P_{51} - P_{out,9}$</td>
<td>$q_{30} + 2q_{36} = q_{42}$</td>
</tr>
<tr>
<td>$L q_{24} = P_{out,1} - P_{24}$</td>
<td>$L q_{60} = P_{72} - P_{60}$</td>
<td>$2q_{43} = 2q_{44} + q_{out,5}$</td>
</tr>
<tr>
<td>$L q_{25} = P_{out,1} - P_{35}$</td>
<td>$L q_{61} = P_{out,8} - P_{58}$</td>
<td>$q_{39} + q_{43} = q_{44} + q_{48}$</td>
</tr>
<tr>
<td>$L q_{26} = P_{out,2} - P_{36}$</td>
<td>$L q_{62} = P_{58} - P_{out,9}$</td>
<td>$q_{39} + q_{44} = q_{45} + q_{49} + q_{out,6}$</td>
</tr>
<tr>
<td>$L q_{27} = P_{out,3} - P_{27}$</td>
<td>$L q_{63} = P_{out,9} - P_{60}$</td>
<td>$q_{40} + q_{45} = q_{46} + q_{60}$</td>
</tr>
<tr>
<td>$L q_{28} = P_{out,4} - P_{28}$</td>
<td>$L q_{64} = P_{38} - P_{44}$</td>
<td>$q_{41} + q_{46} = q_{47} + q_{51} + q_{out,7}$</td>
</tr>
<tr>
<td>$L q_{29} = P_{out,4} - P_{29}$</td>
<td>$L q_{65} = P_{out,9} - P_{55}$</td>
<td>$q_{42} + 2q_{47} = q_{52}$</td>
</tr>
<tr>
<td>$L q_{30} = P_{out,4} - P_{30}$</td>
<td>$L q_{66} = P_{50} - P_{56}$</td>
<td>$2q_{48} = 2q_{53}$</td>
</tr>
<tr>
<td>$L q_{31} = P_{out,5} - P_{35}$</td>
<td>$L q_{67} = P_{34} - P_{55}$</td>
<td>$q_{49} + q_{53} = q_{44} + q_{47}$</td>
</tr>
<tr>
<td>$L q_{32} = P_{out,5} - P_{36}$</td>
<td>$L q_{66} = P_{55} - P_{66}$</td>
<td>$q_{50} + q_{44} = q_{55} + q_{48}$</td>
</tr>
<tr>
<td>$L q_{33} = P_{out,5} - P_{37}$</td>
<td>$L q_{69} = P_{55} - P_{out,10}$</td>
<td>$q_{51} + q_{55} = q_{56} + q_{59}$</td>
</tr>
<tr>
<td>$L q_{34} = P_{out,6} - P_{38}$</td>
<td>$L q_{70} = P_{56} - P_{70}$</td>
<td>$q_{52} + 2q_{56} = q_{60}$</td>
</tr>
<tr>
<td>$L q_{35} = P_{out,6} - P_{39}$</td>
<td>$L q_{71} = P_{out,10} - P_{70}$</td>
<td>$2q_{57} = 2q_{61} + q_{out,8}$</td>
</tr>
<tr>
<td>$L q_{36} = P_{out,5} - P_{30}$</td>
<td>$L q_{72} = P_{70} - P_{72}$</td>
<td>$q_{58} + q_{61} = q_{62} + q_{64}$</td>
</tr>
<tr>
<td>$L q_{37} = P_{out,6} - P_{30}$</td>
<td>$L q_{73} = P_{out,10} - P_{72}$</td>
<td>$q_{59} + q_{62} = q_{63} + q_{65} + q_{out,9}$</td>
</tr>
<tr>
<td>$L q_{38} = P_{out,5} - P_{38}$</td>
<td>$L q_{74} = P_{out,10} - P_{70}$</td>
<td>$q_{60} + 2q_{63} = q_{66}$</td>
</tr>
<tr>
<td>$L q_{39} = P_{out,6} - P_{38}$</td>
<td>$L q_{75} = P_{out,10} - P_{70}$</td>
<td>$2q_{64} = 2q_{71}$</td>
</tr>
<tr>
<td>$L q_{40} = P_{out,7} - P_{40}$</td>
<td>$L q_{76} = P_{out,10} - P_{70}$</td>
<td>$q_{65} + q_{71} = q_{68} + q_{69}$</td>
</tr>
<tr>
<td>$L q_{41} = P_{out,7} - P_{42}$</td>
<td>$L q_{77} = P_{out,10} - P_{70}$</td>
<td>$q_{66} + 2q_{68} = q_{70}$</td>
</tr>
<tr>
<td>$L q_{42} = P_{out,8} - P_{42}$</td>
<td>$L q_{78} = P_{out,10} - P_{70}$</td>
<td>$2q_{69} = 2q_{71} + q_{out,10}$</td>
</tr>
<tr>
<td>$L q_{43} = P_{out,9} - P_{42}$</td>
<td>$L q_{79} = P_{out,10} - P_{70}$</td>
<td>$q_{70} + 2q_{71} = q_{72}$</td>
</tr>
<tr>
<td>$L q_{44} = P_{out,10} - P_{50}$</td>
<td>$L q_{80} = P_{out,10} - P_{72}$</td>
<td>$4q_{72} = 0$</td>
</tr>
</tbody>
</table>
but rather just a few of the many options. Individually, they offer differing levels of reduction. But when multiple levels of reduction are combined, they can become much more powerful. For example, the grid-like network studied lends itself well to a combination of all three methods: a network with inhomogeneous inlet pressures can be divided into simpler subnetworks using the principle of superposition, after which one can then further reduce each subnetwork by taking advantage of similarity axis that exist in the system. Finally, one can then compare results from generation to generation to develop an approximation for much larger networks.

While the above methods are simple in nature, they can still prove to be useful analysis tools for network analysis. And while the previous sections used grid-like
networks as a motivating example, each of these methods have potential use for other network topologies. Additionally, it should be remembered that the numerical analysis of Chapter 4 is equally valid for a grid-like network (it is topology independent) and that analysis can also be applied to aid in analysis - for both the linear and non-linear flow cases. Furthermore, it should also be noted that this chapter is a work in progress; it still remains to fully apply these methods to very large networks. Instead, a framework for reduction has been created and coefficients for various flow parameters have been computed analytically. There
Figure 5.19: The $1/8^{th}$ model of a third generation network used for analysis.
may be a way to do this automatically, and it would even better to do analytically such as was done for the bifurcating tree network, but it was not done here.
Model-based system identification is used for a variety of applications, including heat transfer correlations and controls. While it can be very useful in describing the behavior of a system, its accuracy is dependent on how well the model being used actually describes the system. If a better model is proposed, then system identification can result in an even better description of the systems actual behavior. For true grey-box systems, the model is derived from a reductionist approach in a way that it depends on unknown parameters which can then be estimated by comparing with experimental results. But in most cases, a model is simply assumed and system identification gives the parameters that best describe the observed behavior. These assumed models are often simple in nature, such as first or second order, as they can do a decent job and more complex models result in additional computational complexity.

One way to improve on the typical first- or second-order model is propose simple fractional-order models. Since the proposed models are often nothing more than an educated guess, there is no reason to assume that behavior of a complex system must be of integer-order. Fractional-order models are just as likely to correctly describe behavior and in most cases better describe complex behavior than integer-order models. In Section 6.1, the use of fractional-order models is proposed and compared to similar integer-order models.
6.1 Fractional-order system identification

In Section 2.3, a complex system composed entirely of integer-order components was shown to exhibit fractional-order behavior. This was done using a reductionist approach, where each of the individual components of the system were first modeled and then the coupling between each of these components were used to construct a model for the entire system. The bottom-up approach can be very useful for simple systems and can even be useful for systems containing some form of self-similarity. But for large scale systems without self-similar properties, or for systems in which there exists a large degree of inherent uncertainty, this approach can be difficult and often result in poor mathematical models. In these types of problems, the alternative approach is to use a gray- or black-box approach to developing a mathematical model. In this kind of top-down approach, the internal details and individual components of the system are neglected and a simple model that describes the entire system is sought. This is accomplished by experimentally measuring a system’s response and formulating a mathematical model that describes the observed behavior. In most cases, simple models, such as first- or second-order, are used as a slight percentage error can be preferable to a complex model. In this chapter, it is proposed that fractional-order models have the potential to provide accurate models for at least one class of systems.

6.2 Nearly exponential transitions

Even when an exact fractional-order model cannot be derived, fractional calculus still has uses when studying complex real-world systems [2]. When developing a model of any real system, there are always certain assumptions or simplifications made. For example, when modeling a network of pipes, the flow may be assumed to be laminar and one dimensional, or head losses at junctions and fittings may
be neglected. While the resulting simple model may be useful, it rarely accurately predicts behavior. In order to produce better results, simplifications and approximations must be discarded resulting in a much more complex model. One alternative to increasingly complex models (and exact solutions) is to find simple analytical approximations to experimental results. Simple models can be of use now to suggest general forms of a model in which parameters can be varied to account for assumptions and simplifications until the model accurately describes the observed behavior. This is often done in heat transfer for everything from Nusselt number correlations to heat exchanger analysis.

In the following sections, systems are examined that all exhibit similar behavior. In each case the system at hand undergoes a transition from one steady state to another, in what seems to be a nearly exponential fashion, as the result of a step change in input. Oftentimes when a change like this is seen, the transition is assumed to be either exponential or the sum of several exponentials, both of which result from the solution of integer-order differential equations, and simple first- or second-order mathematical models are used. But it is often the case that the Mittag-Leffler function, which is a generalization of the exponential and arises in the solution of fractional-order differential equations, can more accurately describe the process than the traditional exponential.

As approximate mathematical models for these nearly exponential transitions, the following are proposed:

\[ c_1 \frac{d}{dt} y(t) + c_2 y(t) = u(t), \quad (6.1) \]
\[ c_1 \frac{d^2}{dt^2} y(t) + c_2 \frac{d}{dt} y(t) + c_3 y(t) = u(t), \quad (6.2) \]
\[ c_1 \frac{d^\alpha}{dt^\alpha} y(t) + c_2 y(t) = u(t), \quad (6.3) \]

where the input \( u(t) \) is taken to be a step function. The first-order model (6.1)
with two parameters results in a single exponential. Equation (6.2) is a second-order model which has a solution in the form of the sum of two exponentials, and Equation (6.3) is a variable-order model with a derivative of order $\alpha$, the solution of which is written in terms of the Mittag-Leffler function. The solutions and data approximations resulting from these models are of the form

$$
\begin{align*}
y(t) & = \frac{1}{c_2} - \frac{1}{c_1}e^{-\frac{c_2}{c_1}t}, \\
y(t) & = a + \frac{a}{b(b-c)}e^{-bt} - \frac{a}{c(b-c)}e^{-ct}, \\
y(t) & = \frac{1}{c_1}t^{\alpha}E_{\alpha,\alpha+1}\left(-\frac{c_2}{c_1}t^{\alpha}\right),
\end{align*}
$$

respectively, where $a = \frac{1}{c_1}$, $b = \frac{c_2}{c_1}$, and $c = \frac{c_3}{c_2}$ in (6.5), and the two parameter Mittag-Leffler function is defined as

$$
E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}.
$$

The Mittag-Leffler function arises in the solution of linear fractional-order differential equations in much the same way that exponentials appear in the solution of integer-order differential equations. In fact, the Mittag-Leffler function, which can be computed directly in software packages such as MATLAB, is a more general version of the exponential and can be used to write the solution to integer-order differential equations in addition to fractional-order ones. The opposite case is not true: exponentials can not be used to write the solution of fractional-order differential equations.

Looking at Equations (6.1) and (6.3), it can also be seen that the solution of (6.1) must be the solution of (6.3) for $\alpha = 1$ and the Mittag-Leffler function must reduce to the exponential. This can easily be seen by writing the exponential function $e^z$ as a power series,

$$
e^z = \sum_{k=0}^{\infty} \frac{z^k}{k!},
$$

128
which when compared to the definition of the Mittag-Leffler function, suggests

\[ E_{1,1}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k+1)} \]

\[ = e^z. \]  

Using this, (6.6) can be reduced to (6.4), as

\[ \frac{1}{c_1} tE_{1,2}(-\frac{c_2}{c_1}t) = \frac{1}{c_1} t \sum_{k=0}^{\infty} \frac{(-\frac{c_2}{c_1}t)^k}{\Gamma(k+2)}, \]

\[ = \frac{1}{c_1} t \left( -\frac{c_2}{c_1} \right) \sum_{k=0}^{\infty} \frac{(-\frac{c_2}{c_1}t)^{k+1}}{\Gamma(k+2)}, \]

\[ = -\frac{1}{c_2} \left( e^{-\frac{c_2}{c_1}t} - 1 \right) \]

\[ = \frac{1}{c_2} - \frac{1}{c_2} e^{-\frac{c_2}{c_1}t}. \]  

6.3 Linear systems

Linear systems are unique in that given a set of linear equations, it is often possible to find an analytical solution. But there are cases when finding an exact analytical solution may not be feasible, particularly when the system of equations is very large. In such a case, an approximation to the solution may be sought by numerically or experimentally determining behavior and searching for a best fit analytical model. In the following two subsections, two linear systems with a step response that appears to be nearly exponential are examined and a fractional-order model is used to approximate numerical results.

6.3.1 A linear toy problem

Consider the data shown in Figure 6.1 which was produced by numerically computing a system’s response to a step change in input. Without knowing what system of equations produced the data shown in Figure 6.1, it is almost impossible
to produce a solution that will exactly describe the data generated. However, by looking at the output, it is possible to suggest several forms of an approximate solution which may adequately describe the data. In this case, the data appears to have a near exponential transition from one steady state to another, so we will use the models proposed in Equations (6.1-6.3).

![Figure 6.1: Step response of a black-box system.](image)

For demonstrative purposes, the data in Figure 6.1 is produced using the following system of first-order differential equations:

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_3, \\
\dot{x}_3 &= -6x_1 - 11x_2 - 6x_3 + 6u(t),
\end{align*}
\]

which is simply the third-order differential equation

\[
\frac{d^3 y}{dt^3} + 6\frac{d^2 y}{dt^2} + 11\frac{dy}{dt} + 6y = 6u(t),
\]

written as a system of first-order ODEs. For initial conditions \(x_1(0) = x_2(0) = \)
\[ x_3(0) = 0 \text{ and } u(t) = H(t), \] the solution is found to be

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3
\end{bmatrix} = \begin{bmatrix}
  1 - 3e^{-t} + 3e^{-2t} - e^{-3t} \\
  3e^{-t} - 6e^{-2t} + 3e^{-3t} \\
  -3e^{-t} + 12e^{-2t} - 9e^{-3t}
\end{bmatrix}, \tag{6.19}
\]

and the solution of \( x_1 \), which is also \( y \) in Equation 6.18, is what is shown in Figure 6.1. Knowing the exact solution to be a sum of three exponentials, it should be expected that a first-order model might provide a decent fit, but a second-order model, whose solution contains two exponentials, should provide a better fit. Using the three models in Equations 6.1, 6.2, and 6.3, a non-linear least squares minimization was performed to find the best fit constants for each model. Table 6.1 shows the best fit parameters and associated least squares error of each model and Figure 6.2 shows the corresponding approximations. As is expected, the second-order fit out-performed the first order model. Additionally, the fractional-order model out performs both the first- and second-order models. This suggests that even for relatively simple models, fractional-order approximations may be of some value.

6.3.2 A fractance device

As another example of a linear system, consider the fractance device described in Chapter 2.3. It was previously shown that, for small \( N \), the system of DAE’s resulting from an \( N \)-generational fractance device could be reduced to a single ODE and solved analytically. It was also shown that for \( N = \infty \), an analytical solution was possible that satisfied a fractional differential equation of order \( \alpha = 1/2 \). But for large \( N \), the solution of the system of DAE’s becomes very complex as the degree of the transfer function \( L_N \) grows as \( 2^N - 1 \). Instead of seeking an analytical solution for large \( N \), let us instead seek an approximate solution by numerically
TABLE 6.1

BEST FIT CONSTANTS FOR THE THREE MODELS PROPOSED IN EQUATIONS (6.1-6.3) APPLIED TO THE LINEAR TOY PROBLEM OF EQUATION (6.18); BEST FITS ARE SHOWN IN FIGURE 6.2; $\chi^2$ IS THE LEAST SQUARES ERROR OF THE FIT

<table>
<thead>
<tr>
<th>Model</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$\alpha$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-Order</td>
<td>3.1626</td>
<td>0.2010</td>
<td></td>
<td></td>
<td>6.7900e-04</td>
</tr>
<tr>
<td>Second-Order</td>
<td>0.9929</td>
<td>1.8945</td>
<td>0.9037</td>
<td></td>
<td>7.3911e-05</td>
</tr>
<tr>
<td>Fractional-Order</td>
<td>2.2338</td>
<td>1.7359</td>
<td>1.7130</td>
<td></td>
<td>3.6854e-05</td>
</tr>
</tbody>
</table>

solving the system of equations and using the models proposed in Equations 6.1-6.3.

The numerical solution for the voltage drop, $\Delta v(t)$, across the device for a step change in current, $i(t)$, is shown in Figure 6.3 for devices with $N = 1, 3, 6,$ and 9 generations, where all devices are composed of resistors and capacitors with resistance and capacitance $R = C = 1$. As expected from Chapter 2.3, as $N$ increases, the solution tends to converge to an $N = \infty$ solution, which we know to satisfy Equation 6.3 with $\alpha = 1/2$, $c_1 = 1$, and $c_2 = 0$. Also from Section 2.3, for $N = 1$, the solution is known and satisfies Equation (6.1) with $c_1 = c_2 = 1$. We would then expect the solutions for small $N$ to be described well by (6.1) and the solutions for large $N$ to approach (6.3) with $\alpha = 1/2$.

Using the three models, best fit parameters were determined for $N = 1, 3, 6,$ and 9 generation systems that minimized the least squares error. The best fits for each of the models are shown in Figure 6.4 for each device and the corresponding best fit constants are given in Table 6.2.
Figure 6.2: Step response of Equation (6.18) and the best fit approximations for each model. Solid line with ⬤ is the numerical solution, dashed line marked with △ is first-order model, dash-dot line with □ is second-order, and dotted line marked with + is fractional-order model.

Looking at Figure 6.4 and Table 6.2, it is obvious that the fractional-order model provides the best fit in every case. Even in cases of small devices ($N = 1$ and $N = 3$) shown in Figures 6.4(a) and 6.4(b), where the system behaves as an exact first-order and very near to it, the fractional-order model matches or exceeds the performance of the first-order model. This should be expected as the first-order model is merely a special case of equation (6.3). Likewise, the second-order fit should always perform as well or better than the first-order model, as the first-order model is again a special case of the second-order (with $c_1 = 0$). As can be seen in Table 6.2, however, the second-order model tends to perform slightly worse
Figure 6.3: Step response of an $N$-generational fractance device composed of resistors and capacitors to a step-input in current $i(t)$. Dotted line for $N = 1$ generations, dash-dot line for $N = 3$, dashed line for $N = 6$, and solid line for $N = 9$. 
Figure 6.4: Curve fits of numerically simulated $N$-generational fractance device. Dotted line marked with $\Diamond$ denotes numerically simulated response, dashed line with $\Delta$ is for First Order model, dash-dotted line with $\square$ for Second Order model, and solid line with $+$ for Fractional Order model.
TABLE 6.2

BEST FIT CONSTANTS FOR THE THREE MODELS PROPOSED IN EQUATIONS (6.1-6.2) FOR N = 1, 3, 6, AND 9 GENERATION FRACTANCE DEVICES; χ² IS THE LEAST SQUARES ERROR OF THE FIT

<table>
<thead>
<tr>
<th>Model</th>
<th>c₁</th>
<th>c₂</th>
<th>c₃</th>
<th>α</th>
<th>χ²</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First-Order</td>
<td>1.000</td>
<td>1.000</td>
<td></td>
<td></td>
<td>2.2361e-12</td>
</tr>
<tr>
<td>Second-Order</td>
<td>8.2862e-04</td>
<td>0.9978</td>
<td>1.0032</td>
<td></td>
<td>6.3321e-08</td>
</tr>
<tr>
<td>Fractional-Order</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td></td>
<td>2.2361e-12</td>
</tr>
<tr>
<td>N = 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First-Order</td>
<td>0.4225</td>
<td>0.8858</td>
<td></td>
<td></td>
<td>2.9925e-04</td>
</tr>
<tr>
<td>Second-Order</td>
<td>2.7885e-10</td>
<td>0.4116</td>
<td>0.9028</td>
<td></td>
<td>3.1941e-04</td>
</tr>
<tr>
<td>Fractional-Order</td>
<td>0.6042</td>
<td>0.6077</td>
<td>0.8151</td>
<td></td>
<td>2.4907e-05</td>
</tr>
<tr>
<td>N = 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First-Order</td>
<td>0.3383</td>
<td>0.8761</td>
<td></td>
<td></td>
<td>1.5598e-03</td>
</tr>
<tr>
<td>Second-Order</td>
<td>1.8928e-05</td>
<td>0.3373</td>
<td>0.8771</td>
<td></td>
<td>1.5628e-03</td>
</tr>
<tr>
<td>Fractional-Order</td>
<td>0.7744</td>
<td>0.2753</td>
<td>0.6040</td>
<td></td>
<td>4.5497e-05</td>
</tr>
<tr>
<td>N = 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First-Order</td>
<td>0.3306</td>
<td>0.8762</td>
<td></td>
<td></td>
<td>2.1151e-03</td>
</tr>
<tr>
<td>Second-Order</td>
<td>1.2366e-09</td>
<td>0.3250</td>
<td>0.8843</td>
<td></td>
<td>2.1248e-03</td>
</tr>
<tr>
<td>Fractional-Order</td>
<td>0.93036</td>
<td>0.0821</td>
<td>0.5251</td>
<td></td>
<td>2.2201e-05</td>
</tr>
</tbody>
</table>

than the first-order model, but this is just a result of requiring that c₁ ≠ 0 in the minimization procedure. The important point to note is that adding a second-order term does nothing to improve the fit. This result is not surprising, as it was expected that for larger systems the solution would approach (6.3) with α = 1/2, not α = 2.

6.4 Non-linear systems

As non-linear systems rarely have analytical solutions, the need for good approximate solutions is even more important. In linear systems that exhibit a transition from one steady state to another, exponentials (and simple integer order models)
can offer decent approximations. But in the real world, systems are rarely linear
and trying to approximate solutions using linear models can lead to poor approxi-
mations. While the fractional-order model proposed earlier is still linear, it is able
to model non-linearities better than traditional integer-order linear models. In the
following subsections, two examples involving nonlinear systems will be examined
where fractional-order models prove to be useful.

6.4.1 Non-linear toy problem

In Section 6.3.1, a linear set of equations was used to produce a numerical
data set which was then compared against the proposed models to find the best
approximation. For this example, the equations are altered slightly to produce a
non-linear set of equations and the same models are then used to approximate a
solution. The nonlinear set of equations used is

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_3, \\
\dot{x}_3 &= -6(x_1)^2 - 11x_2 - 6x_3 + 6u(t),
\end{align*}
\]

which can also be written as the non-linear third-order differential equation

\[
\frac{d^3 y}{dt^3} + 6\frac{d^2 y}{dt^2} + 11\frac{dy}{dt} + 6y^2 = 6u(t).
\]

For initial conditions \(x_1(0) = x_2(0) = x_3(0) = 0\) and \(u(t) = H(t)\), the solution was
computed numerically and is shown in Figure 6.5 along with the first, second, and
fractional order approximations. The best fit parameters for each of the models
are shown in Table 6.3. While none of the models perform as well as in the linear
example, the fractional-order model again offers the best fit.
Figure 6.5: Step response of Equation 6.23 and the best solutions using the three proposed models. Dotted line with ⋄ is the numerical solution, dashed line marked with △ is first-order model, dash-dot line with □ is second-order, and solid line marked with + is fractional-order model.
TABLE 6.3

BEST FIT CONSTANTS FOR THE THREE MODELS PROPOSED IN EQUATIONS (6.1-6.3) APPLIED TO THE NON-LINEAR TOY PROBLEM OF EQUATION (6.1-6.3); BEST FITS ARE SHOWN IN FIGURE 6.5; $\chi^2$ IS THE LEAST SQUARES ERROR OF THE FIT

<table>
<thead>
<tr>
<th>Model</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$\alpha$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-Order</td>
<td>2.5483</td>
<td>0.3960</td>
<td></td>
<td></td>
<td>1.5687e-03</td>
</tr>
<tr>
<td>Second-Order</td>
<td>0.8614</td>
<td>1.6737</td>
<td>0.8130</td>
<td></td>
<td>3.3758e-04</td>
</tr>
<tr>
<td>Fractional-Order</td>
<td>2.0193</td>
<td>1.5662</td>
<td>1.7777</td>
<td></td>
<td>3.9169e-05</td>
</tr>
</tbody>
</table>

6.4.2 Experimental results: shell-and-tube heat exchanger

All of the previous examples took advantage of systems where an analytical, or at least numerical, solution was available. In experimental studies, however, it is often the case that a solution is not available. Even in the cases where a dynamic model is available, it is only capable of providing an approximate solution which must still be compared to the experimental results. An analytical solution to the dynamic model is even more rare, particularly for complex systems composed of many equations, possibly non-linear. In these cases, grey- and black-box methods of analysis can be the best way to find a simple, accurate model.

As an example of an experimental case where fractional-order models could be useful, consider a shell-and-tube heat exchanger of the type shown in Figure 6.6. While steady-state correlations for heat exchangers are readily available, transient analysis and correlations are not. In this example, it is desired to develop a dynamic correlation relating the hot-side outlet temperature to a step change in cold-side flow rate while the hot-side flow rate remains constant.
A white-box model of this system is not very practical for this purpose as a model would need to include both axial and radial conduction in the tube, convection along both boundaries of the tube, advection within both fluids, losses to the environment, etc. Additionally, it would need to correctly model turbulent regions and recirculation within the heat exchanger. In short, a white-box model would result in a non-linear system of PDEs (full Navier-Stokes) that would require a numerical solution that, despite its computational cost, still would not accurately describe the process due to the assumptions, such as constant heat transfer coefficients and laminar flow, that would be necessary. Additionally, the solution would be overly complex as only the step-response of the hot-side outlet temperature is desired. A grey-box analysis would be more appropriate, as assumptions such as one-dimensional flow and tube walls could be made to arrive at a simpler mathematical model and experimental results and a fitting of free parameters could help account for the assumptions made. But, by far, the simplest technique is to analyze the heat exchanger system as a black-box and use experimental results to find a correlation between the hot-side outlet temperature and a step change in
cold-side flow rate.

For this experiment, a shell-and-tube heat exchanger was tested and the inlet and outlet temperatures on both the cold and hot side were recorded. The inlet flow rates were held constant in both the cold and hot sides and the inlet temperatures were controlled to remain constant. After the system reached equilibrium for a given set of conditions, a step change (on-off) in the flow rate of the cold-side was applied. Experimental results of two tests are shown in Figure 6.7, where the non-dimensional temperature is given by

$$\theta(t) = \frac{t_{h,o}(t) - t_{h,o}(0)}{t_{h,o}(\infty) - t_{h,o}(0)}, \quad (6.24)$$

and time is non-dimensionalized by

$$\tau = \frac{t}{\tau_r}, \quad (6.25)$$

where \(\tau_r\) is the rise time, defined as the time required for \(t_{h,o}(t)\) to reach 85\% of \(t_{h,o}(\infty)\).

Again, the three models proposed in Equations (6.1-6.3) were used to find best fit approximations. The best fit constants for each model can be seen in Table 6.4 for both data sets and the corresponding approximations are shown in Figure 6.8. In both data sets, the fractional-order model resulted in a better approximation than either the first or second order models. Furthermore, data set #1 is shown to be fit best with a derivative of order \(\alpha = 1.8911\) and data set #2 with a derivative of order \(\alpha = 1.7481\), suggesting that a fractional-order model with a derivative between the two may offer the best fit when applied to the two data sets. The results of taking the average of the two best fit \(\alpha\)'s to give \(\alpha = 1.8196\) and repeating the least squares analysis while holding \(\alpha\) constant in Equation 6.3 are also included in Table 6.4 and shown in Figure 6.9. Looking at the \(\chi^2\) error,
it is noted that the fractional-order model with $\alpha = 1.8196$ outperforms both the first and second order models as well, despite the same number of free parameters.

### TABLE 6.4

**BEST FIT CONSTANTS FOR THE THREE MODELS PROPOSED IN EQUATIONS (6.1-6.2) FOR THE TWO SHELL-AND-TUBE HEAT EXCHANGER DATA SETS. $\chi^2$ IS THE LEAST SQUARES ERROR OF THE FIT WHEN COMPARED TO THE FILTERED DATA**

<table>
<thead>
<tr>
<th>Data Set #1</th>
<th>Model</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$\alpha$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set #1</td>
<td>First-Order</td>
<td>1.0401</td>
<td>0.0286</td>
<td></td>
<td></td>
<td>3.5284e-03</td>
</tr>
<tr>
<td></td>
<td>Second-Order</td>
<td>0.0971</td>
<td>0.5909</td>
<td>0.8997</td>
<td></td>
<td>7.6902e-04</td>
</tr>
<tr>
<td></td>
<td>Fractional-Order</td>
<td>0.2238</td>
<td>2.1700</td>
<td>1.8911</td>
<td></td>
<td>9.5609e-05</td>
</tr>
<tr>
<td></td>
<td>$\alpha = 1.8196$</td>
<td>0.2486</td>
<td>2.0221</td>
<td></td>
<td></td>
<td>1.3355e-04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set #2</th>
<th>Model</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$\alpha$</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First-Order</td>
<td>1.2536</td>
<td>-0.3290</td>
<td></td>
<td></td>
<td>1.4135e-03</td>
</tr>
<tr>
<td></td>
<td>Second-Order</td>
<td>0.1236</td>
<td>0.6504</td>
<td>0.8556</td>
<td></td>
<td>1.0973e-04</td>
</tr>
<tr>
<td></td>
<td>Fractional-Order</td>
<td>0.3318</td>
<td>1.8782</td>
<td>1.7481</td>
<td></td>
<td>2.7757e-05</td>
</tr>
<tr>
<td></td>
<td>$\alpha = 1.8196$</td>
<td>0.2976</td>
<td>2.0476</td>
<td></td>
<td></td>
<td>4.9404e-05</td>
</tr>
</tbody>
</table>

6.5 Fractional-order approximation conclusions

From the examples in this Chapter, it can be seen that fractional-order models and Mittag-Leffler function solutions do have the potential to be of use in the modeling of complex systems. Both linear and non-linear examples have been analyzed using a black-box approach and shown to be better described by linear fractional-order models than traditional integer order linear models. This is not to say that any of these systems truly behave as a fractional-order system, but rather that fractional-order models can produce better approximations than other similar integer order models. In these examples, systems which undergo a nearly exponential transition from one steady state to another were considered. These
Figure 6.7: Raw and filtered outlet temperature step response for a shell-and-tube heat exchanger. Raw data is filtered using a 5th order Savitsky-Golay filter to remove high frequency noise.
Figure 6.8: Best fit approximations to shell-and-tube data using first, second, and fractional-order models. Dotted line is filtered data, dashed line is first-order model, dash-dot is second-order model, and solid line is fractional-order model.
Figure 6.9: Best fit approximations to shell-and-tube data using first-, second-, and fractional-order ($\alpha = 1.8196$) models. Dotted line is filtered data, dashed line is 1st order model, dash-dot is 2nd order model, and solid line is fractional-order model with $\alpha = 1.8196$.

Types of systems are commonly modeled as simple first or second order systems which is why the models used in this Chapter are of the form (6.1-6.3). However, fractional-order models in other forms can be useful for other systems as well, particularly for dynamic heat transfer correlations. Controls is another application where fractional-order models can be useful, as systems are often assumed to be integer order for PID parameter tuning and for various robust and model-based control schemes.
It would be beneficial to remember at this time the motivation for the preceding chapters. In Chapter 1, we considered reduction as it pertains to engineering systems. We noted the variety of methods used in the process of reduction, both in the process of reducing a physical system to a mathematical model and in the process of reducing a mathematical model into a useful form or solution. These methods can be generally classified as belonging to one of three classes: white-, grey-, or black-box reduction methods.

White-box methods are the natural result of the application of methodological reductionism. This form of reductionism can claim Descartes, who believed that complex things could be broken into many smaller, simpler things, as one of its greatest champions. Philosophically, it states that the best way to understand a large system is to first gain a clear understanding of its smallest subsystems or components and their interactions. With this knowledge in hand, the behavior of the larger system can be understood. Reduction from a physical system to a mathematical model in this sense means constructing a model from first principles. For hundreds of years this has been the prevailing method in engineering and science because “it works” [24].

The philosophy behind black-box methods is the diametric opposite to method-
ological reductionism. It finds its roots in mysticism and holism and the idea that one cannot hope to fully understand the behavior of a system based solely on an understanding of its component parts. While this idea seems more modern and natural within the context of complex systems, its roots are much deeper than those of methodological reductionism. It counts among its early champions Aristotle and John Stuart Mill who allowed for the whole to be more than the sum of the parts.

Grey-box methods fall somewhere between the two extremes. Models are postulated based on first principles, but parameters are allowed to permit some degree of flexibility. These parameters are then chosen by finding the best fit between the model and experimental results or observations. Similarly, reduction methods which first study similar, but simpler, models and use the results to postulate approximations for larger and more complex systems can be classified as belonging to the grey-box class.

In addition to surveying the spectrum of reduction methods we also briefly considered the concept of complex systems and the difficulties inherent in modeling them using traditional white- and grey-box approaches. Properties such as size, a high degree of interconnectedness, and emergence serve to make modeling from the bottom up extremely difficult and often leads to intractable problems. Even when a model can be developed from first principles, solving for dynamic solutions can be prohibitively expensive unless additional reductions or approximations can be made. Because of this, black-box methods are commonly used for both simulation and prediction of complex systems. The purpose of this dissertation and the preceding chapters is to offer several methods to aid in the reduction and approximation of complex systems. In particular, self-similar potential driven flow networks were offered as a motivating example of a complex system and white-,
grey-, and black-box methods were presented to aid in reduction.

7.1 Review of work presented

In Chapter 2, systems exhibiting mathematical self-similarity were considered and that similarity was used to aid in reduction. First, an infinite system of linear ODEs was reduced to a single ODE by taking the Laplace transform of the equation set and rearranging such that the resulting power series could be summed to a finite result. Secondly, a PDE was considered. By discretizing the PDE in space, an infinite system of ODEs was produced and the same procedure was used to reduce the model to a single ODE. Of particular note in this example, it was observed that an infinite system of ODEs could be simplified to a single fractional-order ODE.

We next considered a system exhibiting physical self-similarity in Chapter 3 and potential driven bifurcating trees were proposed as an example system. Using a white-box approach to modeling, a mathematical model was developed consisting of a coupled system of differential and algebraic equations. Even for the relatively simple system being considered, solving this DAE system proved to be prohibitively expensive for networks as small as 10 generations. However, by considering the self-similarity in the DAE system it was shown to be possible to reduce the DAE system to a single ODE. For an infinite network the resulting ODE proved to be even simpler in form to determine. It was also shown that as the network grows in size the system’s behavior tends to converge to that of the infinite network. This observation suggests that instead of approximating a large network as a network smaller in scale, it is possible to achieve better results by approximating it as an infinite network. Again it was observed that fractional-order ODEs could appear - even when considering a network composed entirely of integer-order components.
Chapter 4 extended the analysis of Chapter 3 for the specific case of non-linear networks hydraulic piping networks. In this case, the previous analysis was not useful as it depended on the use of Laplace transforms. Instead, a numerical algorithm was proposed for dynamic simulation. In this case, reduction was obtained by taking advantage of the fact that at each time step the velocity field must be divergent free. Using this fact, it was shown to be possible to effectively decouple the computations for the velocity and pressure fields. As before, a mathematical model was derived from a bottom-up analysis and a reduction was performed in the mathematical model to allow for simplification. This method, while used on a bifurcating tree network to demonstrate the numerical algorithm, it is equally applicable to flow networks regardless of topology. In fact, the presented form of the method is specific to pipe flow, but the methodology is also applicable to any potential driven transport network with differential transport equations.

A grid-like network topology was considered in Chapter 5 and several potential methods of reduction were discussed. For a grid-like network, the principle of superposition has potential use for reducing the complexity of a mathematical model. Self-similarity was also used for reduction. For the grid-like network considered, the behavior from one generation to the next was studied as the network increased in size. This can allow one to observe potential patterns and trends in flow quantities which can be used to approximate the behavior of much larger networks. For the example considered this reduction can be considered a grey-box method. Analytical approximations (curve fits) to each flow quantity can be constructed from analyzing smaller networks and then used to approximate larger networks. Last, lines of similarity were used to reduce the mathematical model in complexity by reducing the size of the DAE system. It should also be noted that the numerical analysis of Chapter 4 is equally applicable to grid-like networks and could be used
in combination with all of the additional reduction methods considered.

Finally, a common black-box reduction method was modified based on observations made in the previous chapters. It was noted several times earlier that fractional-order derivatives could result when analyzing complex systems even in cases where the system consists entirely of integer-order components. Using this observation, a fractional-order model was used to best fit several nearly-exponential transitions resulting from both linear and non-linear toy problems. It was shown that even for these toy problems, the fractional-order system model provided better curve fits than the traditional first- and second-order models. Furthermore, experimental data was collected for a shell-and-tube heat exchanger and analyzed using the same three models. Again, the fractional-order model provided the best fit. While the claim is not being made that the heat exchanger actually behaves as a fractional-order system, it is noteworthy that the fractional model provides the best fit (despite fewer fit parameters than the 2nd order linear model). As the solution to a the fractional-order model can be written in terms of the Mittag-Leffler function which was shown to be closely related to a series of exponentials, perhaps this should not be so surprising.

7.2 Indirect implications of presented work

In addition to the results previously presented, there are several indirect implications that should be noted and considered. First, in the analysis of Chapter 2 and Chapter 3, self-similarity was used to reduce mathematical models. In the context presented, mathematical models of large, or even infinite, physical systems were reduced to a single ODE. This reduction should not be overlooked simply as the reduction of a particular physical system. It could be presented in another context as the reduction of a large or even infinite system of ODEs, or even a
system of DAEs, to a single ODE.

It was also shown in Chapters 2 and 3 that a system composed entirely of integer-order components can result in a fractional-order ODE. Fractional-order ODEs are not a new thing; they have been studied by mathematicians since the days of Leibniz and L’Hôpital. But the analysis is always in terms of the application of the operator

\[ D = \frac{d}{dt}, \quad (7.1) \]

a non-integer (i.e., fractional) number of times.

The examples presented in Chapter 3 demonstrate this in that the operators being used are integer-order integro-differential equations with respect to time. However, the analysis is intentionally left generic enough to be valid for any linear operator - be it algebraic, integro-differential with respect time time, matrix, etc.. This suggests a generalization of fractional-order derivatives to fractional-order operators; that fractional-order derivatives are merely a subset of the possible fractional-order operators. For example, while

\[ (D)^{1/2} = \frac{d^{1/2}}{dt^{1/2}}, \quad (7.2) \]

is well defined and understood within the context of fractional-derivatives, expressions such as

\[ (D + 1)^{1/2} = \left(\frac{d}{dt} + 1\right)^{1/2}, \quad (7.3) \]

is not. Continuing with the example of the bifurcating tree piping network of Chapter 3, asymmetric branching could give rise to an operator of the form

\[ L = \left( (D + \alpha)(D + \beta) \right)^{1/2}. \quad (7.4) \]

What these operator mean are not understood at this time, but future work should be focused to gain a better understanding of it.
There is also an important observation made about approximating large systems. Approximations were made for the self-similar bifurcating tree network considered in earlier chapters in two ways: first, consider that observations made about small-scale networks were used as an approximation for larger networks. This is the typical approach when analyzing large scale systems. A smaller model of the network is considered and those results are extended or extrapolated to predict or approximate the behavior of a larger network on which the same analysis cannot be easily applied. Secondly, and more remarkable, is the case of approximating a large network by an even larger, or possibly even infinite, network. This runs counter in many ways to the idea of reduction: where one typically reduces a complex item or system to one that is less complex in nature. But it was shown in the case of bifurcating networks that it is actually simpler to reduce an infinite network than it is to an extremely large one. In this sense, one is reducing a complex item to a simpler one by approximating it as one even more complex.

7.3 Recommendations for future work

Present work has so far centered on reducing overly complex mathematical models resulting from a reductionist approach to the modeling of large self-similar potential driven flow systems, and the improvement of black-box system identification by using simple fractional-order models. Future work could continue to focus on these areas.

There are many forms of self-similarity seen in science and engineering. While bifurcating tree networks are a very common occurrence, particularly in biological and engineering systems, there are other self-similar networks that occur naturally and with great frequency, such as the grid-like network briefly presented in Chapter 5. There is ample room to continue the development started, both with the
methods presented and for the development of new methods. In particular, work should continue toward the development of an analytical approximation for \( N \) and infinite generation networks such as that developed for the bifurcating tree. Additionally, Group Theory, Lie Groups, and similar approaches to similarity reduction are very much on the fringes of everything discussed within this dissertation. Their relationships, however, are somewhat nebulous, but I believe there must be a way to bring them together.

In a similar manner, there are many open mathematical questions in regards to convergence in the frequency domain of the continued fractions that were developed in the reduction of self-similar systems. While a detailed analysis of this convergence was no considered here, it is noted that it is of great interest. What convergence means in this space should be further studied as it is not obvious. How the Laplace variable \( s \) is thought of plays a large roll in this, as it, as a polynomial is a bounded operator, while the differential operator itself is unbounded. Defining a norm in this space is thus somewhat difficult. These are all issues that should be considered in future development and could have very interesting results.

Furthermore, the present work on bifurcating trees can be extended to analyze disproportionate branching as a model of arterial/capillary distribution. For highly disproportionate branching, the solution should be a perturbation of regular pipe flow. Networks with stochastic or probabilistic branching could also be examined. The addition of a stochastic or probabilistic component to the piping network could yield very interesting results as it has already been been shown that fractional-order derivatives can result from stochastic processes in the same way that we have seen it appear in self-similar networks [10]. In addition to further development of the analysis of both tree networks and grid-like networks, there are a number of other network topologies that could offer interesting results.
The reduction of PDEs is closely tied to the reduction of transfer networks, as a discretized PDE can often be represented as a transfer network. This was examined briefly in Chapter 2. In the same way that the one dimensional heat equation was discretized and analyzed, the two-dimensional heat equation can also be considered. The grid-like network should be a natural parallel to this and should aid as in the development. Additionally, other PDEs can be reduced to give boundary condition transformations. Pennes equation or the fin equation, are examples of PDEs where such a boundary condition transformation would have practical applications such as in laser or cryogenic surgery [32,57,60]. When using cryogen sprays to locally destroy tissue in a procedure such as laser removal of port wine birth marks as in Figure 7.1 or for cancer treatment, it is vitally important that the amount of heat removed from the tissue be carefully monitored so as not to destroy the healthy tissue surrounding the affected tissue. So a formula such as (2.46) relating the surface heat flux and the surface temperature, which can be easily measured, is of great value.

![Cryogen Spray Treatment](image)

Figure 7.1: Cryogen spray treatment of affected tissue.

In the previous section, it was already mentioned that future work could be concentrated on the study of fractional-order operators.
As for the use of fractional-order models in system identification problems, work could proceed in two directions. First, the development of dynamic correlations for heat transfer can be improved and the advantages of fractional-order models can be more clearly shown by using more data and finding the models that produce the best fit for all data sets as opposed to individual tests. Other functions, such as the stretched exponential [52], could be used in addition to the Mittag-Leffler function for better correlations.

Secondly, fractional-order models can be used for control purposes. It has been shown that fractional-order systems, which are merely systems better described by fractional-order models, can be controlled more efficiently using fractional-order controllers, such as Podlubny’s $PI^\lambda D^\mu$ controller, than by using traditional integer-order controllers [47]. As a fractional-order model can been identified using fractional-order system identification, a method or formula for choosing the appropriate $\lambda$, $\mu$, and controller gains, much like the often used Ziegler-Nichols formulas for $PID$ controllers [66], can be developed for $PI^\lambda D^\mu$ control. Implementation of such a method can be tested in an application such as temperature control in a plane wall.

While this dissertation offered several possible methods of reduction for large-scale self-similar networks, it should be obvious that there exists a great deal of room for future work - both directly and indirectly related to the current work presented.
APPENDIX A

NUMERICAL IMPLEMENTATION OF INTEGER AND FRACTIONAL ORDER OPERATORS

This appendix is intended as a short survey of how fractional-order operators can be derived and implemented numerically. Its purpose is to aid the curious reader with an understanding of how fractional-order derivatives and fractional-order integrals relate to their integer-order counterparts.

In the following
• $h, \alpha, \beta \in \mathbb{R}$
• $i, j, m, n \in \mathbb{N} = \{1, 2, 3, \ldots\}$
• $s \in \mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
• $f, g : \mathbb{R} \mapsto \mathbb{R}$ are sufficiently smooth functions of $t$
• $I$ is the identity operator

A.1 Forward shift operator

Definitions

\[
E_h f \overset{\text{def}}{=} f(t + h) \quad (A.1)
\]

\[
E_h^n f \overset{\text{def}}{=} \underbrace{E_h (E_h \ldots (E_h f) \ldots)}_{n \text{ times}} \quad (A.2)
\]

\[
(E_h^n + E_h^m) f \overset{\text{def}}{=} E_h^n f + E_h^m f \quad (A.3)
\]
Properties

\[ E[n](f + g) = E[n]f + E[n]g \]  
\[ E[n](\alpha f) = \alpha E[n]f \]  
\[ E_{nh} f = f(t + \alpha h) \]  
\[ E[n] f = f(t + nh) \]  
\[ E_h(fg) = E_h f E_h g \]  
\[ E^n_h = E_h \]  
\[ E_0 = I \]  
\[ E^n_h = E_{nh} \]  
\[ E[n] E[m] = E[n+m] \]  
\[ (E[n]_{h})^m = E[m]_{h} \]  
\[ (E[n]_{h} + E[m]_{h})^j = \sum_{i=1}^{j+1} \frac{j!}{(i-1)!(j-i+1)!} E^{n(i-1)}_{h} E^{m(j-i+1)}_{h} \]

where two operators are equivalent if, when operating on the same arbitrary function, the two operators produce the same result.

Extension to negative integers

A forward shift operator taken to a positive power, \( E^n \), results in a forward shift. A forward shift operator taken to a negative power, \( E^s \), where \( s = -n \), results in a backwards shift. Thus, the backward shift operator of order \( m \) is merely the forward shift operator raised to the power \(-m\).

\[ E^n_{sh} f \overset{\text{def}}{=} f(t + sh) \]  
\[ E^n_{h} = E_{sh} \]  
\[ E^0_{h} = I \]
Extension to reals

\[ E_h^\alpha f \overset{\text{def}}{=} f(t + \alpha h) \quad (A.18) \]

\[ E_h^\alpha = E_{\alpha h} \quad (A.19) \]

\[ E_h^\alpha E_h^\beta = E_h^{\alpha + \beta} \quad (A.20) \]

Inverse operators

If \( Y \) is a function for which application of the shift operator \( E_h^1 \) yields \( y(x) \), then we can write \( E_h^{-1} y = Y \), i.e.,

\[ \text{if } E_h^1 Y(x) = y(x), \text{ then } E_h^{-1} y(x) = Y(x). \quad (A.21) \]

Then the inverse, \( E_h^{\beta} \), of an operator \( E_h^\alpha \) is defined as \( E_h^{\beta} = E_h^{-\alpha} \), where \( \beta = -\alpha \), so that \( E_h^\beta E_h^\alpha = E_h^0 = I \) are all equivalent operators.

The inverse of the Shift operator \( E_h^\alpha \), for \( \alpha \in \mathbb{R} \), always exists and is unique and can be easily shown to be \( E_h^{-\alpha} \):

\[ \text{if } E_h^{-\alpha} f(x) = f(x - \alpha h), \text{ then } E_h^{-\alpha} E_h^\alpha f(x) = E_h^{-\alpha} [f(x + \alpha h)] = f(x). \quad (A.22) \]

The inverse of the forward shift operator \( E_h^\alpha \) is the backward shift operator \( E_h^{-\alpha} \).

Null space of the shift operator

The null space, \( \mathcal{N}(L) \), of a linear operator \( L : X \rightarrow Y \) is the subset of \( X \) defined by \( \mathcal{N}(L) = \{ x \in X : Lx = 0 \} \). Taking \( L = E_h^\alpha \), the only \( x \in \mathcal{N}(E_h^\alpha) \) is the origin of \( X \), that is the trivial solution \( x = 0 \). For the shift operator then, the null space is trivial, that is \( \mathcal{N}(E_h^\alpha) = \{0\} \).
Numerical implementation

Numerical implementation of the forward shift operator $E^s_h$ is straightforward. For the operator $E^s_h$, where $s \in \mathbb{Z}$ and the function $F(t)$ is sampled such that $F_i = F(t_0 + ih)$, then

$$E^s_h F_i = F(t_0 + ih + sh)$$
$$= F(t_0 + (i + s)h)$$
$$= F_{i+s}$$ \hspace{1cm} (A.23)

Implementation of $E^\alpha_h$ requires that the function be sampled at intervals no greater than $\alpha h$. Given a signal $F(t)$ and the freedom to sample it at any time to produce the data set $F_i = F(t_0 + i\alpha h)$, where $i = 1, 2, 3 \ldots$, then

$$E^\alpha_h F_i = F(t_0 + i\alpha h + \alpha h)$$
$$= F(t_0 + (i + 1)\alpha h)$$
$$= F_{i+1}$$ \hspace{1cm} (A.26)

Linear equations

To find $f(t)$ given $g$ and

$$Lf = g$$ \hspace{1cm} (A.29)

where

$$L = \sum_{i=1}^{j} p_i(x) E^\alpha_i,$$ \hspace{1cm} (A.30)

and $p_i(x)$ and $g(x)$ are real valued functions and $p_i(x) \neq 0$ for all $x \geq x_0$, then the initial conditions

$$E^\alpha_i f = a_i, \quad \text{for } i = 1, 2, \ldots, j - 1,$$ \hspace{1cm} (A.31)

are necessary for a unique solution. Furthermore, the solution procedure is similar to the solution of ordinary linear differential equations. The fundamental solution
set can be found by substituting $\lambda^n$ for $x$ into the homogeneous equation and solving the resulting characteristic equation. Particular solutions can be found using an analog of the method of undetermined coefficients or variation of parameters.

A.2 Forward difference operator

Definitions

\[ \Delta_h f \overset{\text{def}}{=} f(t + h) - f(t) \tag{A.32} \]

\[ \Delta^n_h f \overset{\text{def}}{=} \underbrace{\Delta_h \left( \Delta_h \cdots \left( \Delta_h f \right) \cdots \right)}_{\text{n times}} \tag{A.33} \]

Properties

\[ \Delta^n_h (f + g) = \Delta^n_h f + \Delta^n_h g \tag{A.34} \]

\[ \Delta^n_h (\alpha f) = \alpha \Delta^n_h f \tag{A.35} \]

\[ \Delta(fg) = f \Delta g + E_g \Delta f = E f \Delta g + g \Delta f \tag{A.36} \]

\[ (\Delta^n_h + \Delta^n_m) f = \Delta^n_h f + \Delta^n_m f \tag{A.37} \]

\[ \Delta^n_h \Delta^n_m f = \Delta^{n+m}_h f \tag{A.38} \]

\[ \Delta_h = E_h - E^0_h \tag{A.39} \]

\[ E_h \Delta_h = \Delta_h E_h \tag{A.40} \]

\[ \Delta^1_h = \Delta_h \tag{A.41} \]

\[ \Delta^n_h = (E_h - I)^n \tag{A.42} \]

\[ = \sum_{k=0}^{n} \binom{n}{k} (-1)^k E^{n-k} \tag{A.43} \]

\[ = \sum_{k=0}^{n} \binom{n}{k} (-1)^{n-k} E^k \tag{A.44} \]
\[
= \sum_{k=0}^{n} (-1)^{n-k} \frac{n!}{k!(n-k)!} E_h^k
\] (A.45)

Extension to negative integers

\[
\Delta_h^0 = I \tag{A.46}
\]
\[
\Delta_h^s = (E_h - E_h^0)^s \tag{A.47}
\]
\[
= \sum_{k=0}^{\infty} \binom{s}{k} (-1)^k E^{s-k} \tag{A.48}
\]
\[
= \sum_{k=0}^{\infty} \binom{s}{k} (-1)^{s-k} E^k \tag{A.49}
\]
\[
= \sum_{k=0}^{\infty} \frac{s(s-1)(s-2)\ldots(s-k+1)}{k!} \frac{1}{(-1)^{s-k}} E_h^k \tag{A.50}
\]

Extension to reals

\[
\Delta_h^\alpha = (E_h - E_h^0)^\alpha \tag{A.51}
\]
\[
\Delta_h^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^k E^{\alpha-k} \tag{A.52}
\]
\[
= \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^{\alpha-k} E^k \tag{A.53}
\]
\[
= \sum_{k=0}^{\infty} \frac{\alpha(\alpha-1)(\alpha-2)\ldots(\alpha-k+1)}{k!} (-1)^{k} E_h^{\alpha-k} \tag{A.54}
\]

Null space of the forward difference operator

The null space of the forward difference operator \( \mathcal{N}\{\Delta_h^\alpha\} \) includes every function \( f \) such that \( \Delta_h^\alpha f = 0 \). Functions that exist in \( \mathcal{N}\{\Delta_h^\alpha\} \) include every periodic function with period \( h \) and for \( \alpha = n \), all polynomials of degree \( n - 1 \). This might
not be a complete list, but the point is that $\mathcal{N}\{\Delta^\alpha_h\}$ is not trivial. For the sake of this summary, $p(x)$ is used to represent the set of functions that exist in $\mathcal{N}\{\Delta^\alpha_h\}$.

Note that even though $\Delta^\alpha_h$ can be written in terms of the shift operator $E^\alpha_h$ and $\mathcal{N}\{E^\alpha_h\}$ is trivial, the null space of the forward difference operator is not trivial.

Inverse operators

If $Y$ is a function whose first difference is the function $y$, then $Y$ is called an indefinite sum of $y$ and is denoted by $\Delta^{-1}_h y$, i.e.,

$$\text{if } \Delta_h Y = y(x), \text{ then } \Delta^{-1}_h y(x) = Y(x) + p(x). \quad \text{(A.55)}$$

where $p(x) \in \mathcal{N}\{\Delta^\alpha_h\}$.

Finding an indefinite sum of $y$ is the problem inverse to finding the difference of $y$. Instead of starting with $y$ and differencing it, we instead seek another function which results in $y$ upon being differenced. If $s$ is a negative integer such that $s = -n$, then the following is true:

$$\Delta^s_h \Delta^n_h f(t) = f(t) \quad \text{(A.56)}$$

$$\Delta^s_h \Delta^n_h f(t) = f(t) + p(x) \quad \text{(A.57)}$$

Similarly,

$$\Delta^\alpha_h \Delta^{-\alpha}_h f(t) = f(t) \quad \text{(A.58)}$$

$$\Delta^{-\alpha}_h \Delta^\alpha_h f(t) = f(t) + p(x) \quad \text{(A.59)}$$

Numerical implementation

There are two possible formulas for implementing the forward difference operator numerically,

$$\Delta^\alpha_h = \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^k E^{\alpha-k} \quad \text{(A.60)}$$
and
\[ \Delta^\alpha_h = \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^{\alpha-k} E^k \]  
(A.61)

The first option results in real coefficients and fractional step sizes, while the second provides complex coefficients with integer order steps (if \( \alpha \notin \mathbb{N} \)). While both are equally valid, the first will be used for convenience sake to avoid the use of complex coefficients. Using this formula, implementation of \( \Delta^\alpha_h \) is straightforward for \( \alpha \geq 0, \in \mathbb{Q} \), although it has the disadvantage of requiring data at fractional step sizes if \( \alpha \notin \mathbb{N} \). However, if \( \alpha \in \mathbb{Q} \) then \( \alpha \) can be written as \( \alpha = \frac{p}{q} \) and the problem of requiring data at fractional time steps can be avoided by taking the sampling rate to be \( f = \frac{q}{h} \). Now given a signal \( F(t) \) and sampling at the appropriate frequency results in the data set \( F_i = F(t_0 + \frac{ih}{q}) \). The forward difference of order \( \alpha \) at point \( F_i \) is now given by
\[ \Delta^\alpha_h F_i = \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^{\alpha-k} E^\alpha_{h} F_i \]  
(A.62)

where
\[ E^\alpha_{h} F_i = F_{i+p-qk}, \quad \alpha = \frac{p}{q} \]  
(A.63)

For \( \alpha \in \mathbb{N} \), this is very simple as every \( \binom{\alpha}{k} = 0 \) for \( k > \alpha \). However, for \( \alpha \geq 0 \) and \( \alpha \notin \mathbb{N} \), \( \binom{\alpha}{k} \neq 0 \) for \( k > \alpha \). Luckily, \( \binom{\alpha}{k} \to 0 \) as \( k \to \infty \), so only a few terms for \( k > \alpha \) are needed to approximate the sum. Also note that the data required to calculate the fractional forward difference at \( F_i \) requires data not only at fractional multiples of \( h \), but also at points both before and after \( t_i \).

For \( \alpha \notin \mathbb{Q} \), numerical implementation of the forward difference operator can only be approximated.

Implementation of \( \Delta^\alpha_h \) is more complicated for \( \alpha < 0 \) because the binomial coefficient does not necessarily tend to zero as \( k \) goes to infinity. While it is
straightforward, there is no guarantee the series will converge which decreases its usefulness.

Implementing an infinite sum numerically is no easy task, and only taking a few terms is not always a good idea because the number of terms needed to be even somewhat accurate depends on \( \alpha \). However, a simple assumption can simplify the infinite sum to a finite sum. When dealing with a discrete data set \( f_i = f(a + \frac{ih}{q}) \), there is always an initial and final data point, \( f_0 = f(a) \) and \( f_n = f(a + n\frac{h}{q}) \). When considering \( \Delta_h^\alpha F_i \) it is obvious to see data is required at points \( F_{i+p-kq} \), which, as \( k \to \infty \), requires data before \( f_0 = f(a) \), which does not exist, or you do not have. However, by making the assumption that \( F_i = 0 \) for \( i < 0 \) (\( F(a+t) \) is causal), the infinite sum can be simplified to a finite sum from \( k = 0 \) to \( k = n \), where \( n = \frac{t-a}{h} \).

Denoting this operator with the causal assumption as \( a \Delta_h^\alpha \), we can write

\[
a \Delta_h^\alpha F_i = \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E_h^{\alpha-k} F_i \quad (A.64)
\]

where

\[
E_h^{\alpha-k} F_i = F_{i+p-kq}, \quad \alpha = \frac{p}{q} \quad (A.65)
\]

and \( n = \frac{t-a}{h} \).

**Linear equations**

To find \( f(t) \) given \( g \) and

\[
\sum_{i=1}^{j} \Delta_h^\alpha i f = g \quad (A.66)
\]

along with \( j \) independent conditions on \( f \).
A.3 Backward difference operator

Definitions

\[
\nabla_h f \overset{\text{def}}{=} f(t) - f(t - h) \tag{A.67}
\]

\[
\nabla^n_h f \overset{\text{def}}{=} \underbrace{\nabla_h (\nabla_h \ldots (\nabla_h f) \ldots)}_{n \text{ times}} \tag{A.68}
\]

Properties

\[
\nabla^n_h (f + g) = \nabla^n_h f + \nabla^n_h g \tag{A.69}
\]

\[
\nabla^n_h (\alpha f) = \alpha \nabla^n_h f \tag{A.70}
\]

\[
\nabla(fg) = f \nabla g + E^{-1} g \nabla f = E^{-1} f \nabla g + g \nabla f \tag{A.71}
\]

\[
(\nabla^n_h + \nabla^m_h) f = \nabla^n_h f + \nabla^m_h f \tag{A.72}
\]

\[
\nabla^n_h \nabla^m_h f = \nabla^{m+n}_h f \tag{A.73}
\]

\[
\nabla_h = E^0_h - E^{-1}_h \tag{A.74}
\]

\[
\nabla_h = E^{-1}_h \Delta_h \tag{A.75}
\]

\[
\nabla_h E_h = E_h \nabla_h \tag{A.76}
\]

\[
\nabla_h = \Delta_h \tag{A.77}
\]

\[
\nabla^1_h = \nabla_h \tag{A.78}
\]

\[
\nabla^n_h = (I - E^{-1}_h)^n \tag{A.79}
\]

\[
\nabla^n_h = \sum_{k=0}^{n} \binom{n}{k} (-1)^{n-k} E_h^{k-n} \tag{A.80}
\]

\[
\nabla^n_h = \sum_{k=0}^{n} \binom{n}{k} (-1)^k E_h^{-k} \tag{A.81}
\]
Extension to negative integers

\[ \nabla_h^0 = I \quad (A.82) \]
\[ \nabla_h^s = (I - E_h^{-1})^s \quad (A.83) \]
\[ = \sum_{k=0}^{\infty} \binom{s}{k} (-1)^{s-k} E_h^{-k} \quad (A.84) \]
\[ = \sum_{k=0}^{\infty} \binom{s}{k} (-1)^{k} E_h^{-k} \quad (A.85) \]

Extension to reals

\[ \nabla_h^\alpha = (I - E_h^{-1})^\alpha \quad (A.86) \]
\[ = \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^{\alpha-k} E_h^{-k} \quad (A.87) \]
\[ = \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^k E_h^{-k} \quad (A.88) \]
\[ = \sum_{k=0}^{\infty} \frac{\alpha(\alpha - 1)(\alpha - 2)\ldots(\alpha - k + 1)}{k!} (-1)^k E_h^{-k} \quad (A.89) \]

Null space of the backward difference operator

The null space of the backward difference operator \( \mathcal{N}\{\nabla_h^\alpha\} \) includes every function \( f \) such that \( \nabla_h^\alpha f = 0 \). Functions that exist in \( \mathcal{N}\{\nabla_h^\alpha\} \) include every periodic function with period \( h \), and for \( \alpha = n \), all polynomials of degree \( n - 1 \). Not surprisingly, any function \( f \in \mathcal{N}\{\nabla_h^\alpha\} \) is also in \( \mathcal{N}\{\Delta_h^\alpha\} \), thus \( p(x) \) is again used to represent the set of all functions that span \( \mathcal{N}\{\nabla_h^\alpha\} \).
Inverse operators

If $Y$ is a function whose first difference is the function $y$, then $Y$ is called an indefinite sum of $y$ and is denoted by $\nabla^{-1}_h y$, i.e.,

$$\text{if } \nabla_h Y = y(x), \text{ then } \nabla^{-1}_h y(x) = Y(x) + p(x). \quad (A.90)$$

where $p(x)$ is the set of functions that span $\mathcal{N}\{\nabla^n_h\}$.

Finding an indefinite sum of $y$ is the problem inverse to finding the difference of $y$. Instead of starting with $y$ and differencing it, we instead seek another function which results in $y$ upon being differenced. If $s$ is a negative integer such that $s = -n$, then the following is true:

$$\nabla^n_h \nabla^s_h f(t) = f(t) \quad (A.91)$$

$$\nabla^s_h \nabla^n_h f(t) = f(t) + p(x) \quad (A.92)$$

Likewise, for $\alpha > 0$,

$$\nabla^n_h \nabla^{-\alpha}_h f(t) = f(t) \quad (A.93)$$

$$\nabla^{-\alpha}_h \nabla^n_h f(t) = f(t) + p(x) \quad (A.94)$$

Numerical implementation

Numerical implementation of the backward difference operator $\nabla^n_h$ for $\alpha \geq 0$ is similar to that of $\Delta^n_h$, but is even more straightforward. Given a function $F(t)$ sampled such that the discrete data set is given by $F_i = F(a + ih)$, then the backward difference of order $\alpha$ at $F_i$ is given by

$$\nabla^n_h F_i = \sum_{k=0}^{\infty} \left( \begin{array}{c} \alpha \\ k \end{array} \right) (-1)^k E^{-k}_h F_i \quad (A.95)$$

where

$$E^{-k}_h F_i = F_{i-k} \quad (A.96)$$
Once again, for $\alpha \in \mathbb{N}$, this is very simple as every ${\alpha \choose k} = 0$ for $k > \alpha$. However, for $\alpha \geq 0$ and $\alpha \not\in \mathbb{N}$, ${\alpha \choose k} \neq 0$ for $k > \alpha$. Luckily, ${\alpha \choose k} \to 0$ as $k \to \infty$, so only a few terms for $k > \alpha$ are needed to approximate the sum. Also note that to calculate the fractional backward difference at $F_i$ requires data only at or before $t_i$ and only at integer step sizes, which are the two main advantages of using the backward difference.

When dealing with a discrete data set $f_i$, there is always an initial data point $f_0 = f(a)$. From the formula above for $\nabla_h^\alpha F_i$, it is easy to see that the numerical implementation of $\nabla_h^\alpha$ requires data at $F_{i-k}$, so as $k \to \infty$, $\nabla_h^\alpha$ requires data before $f_0$, which does not exist. However, by assuming $F_i$ to be causal ($F_i = 0$ for $i < 0$), the formula can be simplified and the infinite sum becomes finite, which can now be easily calculated. Denoting $a\nabla_h^\alpha$ as the backward difference operator with $F_i = 0$ for $i < 0$, we can write

$$a\nabla_h^\alpha F_i = \sum_{k=0}^{n} \left(\frac{\alpha}{k}\right) (-1)^k E_h^{-k} F_i$$  \hspace{1cm} (A.97)

where

$$E_h^{-k} F_i = F_{i-k}$$  \hspace{1cm} (A.98)

and $n = \frac{t-a}{h}$.

Implementation of $\nabla_h^\alpha$ is more complicated for $\alpha < 0$ because the binomial coefficient does not necessarily tend to zero as $k$ goes to infinity. While it is straightforward, there is no guarantee the series will converge which decreases its usefulness.

Alternatively, as was the case with the forward difference, there is another way to implement the backward difference operator that has decided disadvantages
when compared to the previous method. The backward difference of a function at point \( F_i \) can also be expressed as

\[
\nabla_h^\alpha F_i = \sum_{k=0}^{\infty} \binom{s}{k} (-1)^{\alpha-k} E_h^{\alpha-k} F_i \tag{A.99}
\]

Now, for \( \alpha \not\in \mathbb{N} \), it is obvious that this implementation results in both imaginary coefficients and fractional step sizes. The combination of the two make it very impractical.

A.4 Differential operator

Definitions

The differential operator can be defined using both the forward and backward difference operators \( \Delta \) and \( \nabla \). With the forward difference operator

\[
D f \overset{\text{def}}{=} \lim_{h \to 0} \left( \frac{\Delta_h f}{h} \right)^n \tag{A.100}
\]

\[
D^n f \overset{\text{def}}{=} D \left( D \ldots (Df) \ldots \right) \tag{A.101}
\]

\[
D^n f = \lim_{h \to 0} \left( \frac{\Delta^n_h f}{h^n} \right) \tag{A.102}
\]

Using the backward difference operator

\[
D f \overset{\text{def}}{=} \lim_{h \to 0} \left( \frac{\nabla_h f}{h} \right)^n \tag{A.103}
\]

\[
D^n f \overset{\text{def}}{=} D \left( D \ldots (Df) \ldots \right) \tag{A.104}
\]

\[
D^n f = \lim_{h \to 0} \left( \frac{\nabla^n_h f}{h^n} \right) \tag{A.105}
\]
Properties

Properties of the differential operator are independent of the choice of difference operator used to define it. The following properties are true for both definitions.

\[(D^n + D^m) f = D^n f + D^m f\]  \hspace{1cm} (A.106)
\[D^n(f + g) = D^n f + D^n g\]  \hspace{1cm} (A.107)
\[D^n(\alpha f) = \alpha D^n f\]  \hspace{1cm} (A.108)
\[D(fg) = f Dg + g Df\]  \hspace{1cm} (A.109)
\[D^1 = D\]  \hspace{1cm} (A.110)
\[D^mD^n f = D^{m+n} f\]  \hspace{1cm} (A.111)

Extension to reals

Like the shift and difference operators, the differential operator can also be extended to include derivatives of arbitrary real order. Using the forward difference definition and \(\alpha > 0\):

\[D^\alpha f = \lim_{h \to 0} \left( \frac{\Delta_h^\alpha f}{h^\alpha} \right)\]  \hspace{1cm} (A.112)

For the backward difference definition:

\[D^\alpha f = \lim_{h \to 0} \left( \frac{\nabla_h^\alpha f}{h^\alpha} \right)\]  \hspace{1cm} (A.113)

Derivatives of order \(\alpha \leq 0\) are not considered in this section, but will be discussed with integral operators.

Null space of the differential operator

The differential operator is another operator with a non-trivial null space. In general, \(\mathcal{N}\{D^\alpha\} = \{f \in F : D^\alpha f = 0\}\). For \(\alpha = n\) where \(n\) is an integer, \(\mathcal{N}\{D^n\} = \)
$C(x)$, the set of polynomial functions of order $n - 1$. If the operator $L$ is taken to be a linear combination of differential operators such as $L = (D^2 + D^1 + D^0)$, then the null space includes functions of the type $f = e^{\lambda x}$. The point, once again, is that the null space of the differential operator is non-trivial.

When solving differential equations of the type $Lf = g$, the homogeneous solution (the solution to the equation $Lf = 0$) is exactly the set of functions (and using the superposition principle, the sum of all such functions) that lie in the null space of the operator $L$.

**Inverse operators**

If $Y$ is a function that when operated upon by $D^\alpha (\alpha > 0)$ results in the new function $y$, then $Y$ is called the antiderivative, or integral of $y$ and is denoted by $D^{-\alpha}y$, i.e.,

$$
if D^\alpha Y(x) = y(x), \text{ then } D^{-\alpha}y(x) = Y(x) + C
$$

(A.114)

The inverse for a differential operator of order $\alpha$ is an integral, or antiderivative, of the same order. However, this inverse is only unique to a constant if initial conditions or boundary conditions are not provided. Likewise, the inverse of an integral operator is a derivative of the same order. This inverse is unique. Mathematically, this can be expressed as

$$
D^1 J^1 f = f
$$

(A.115)

$$
J^1 D^1 f = f + C
$$

(A.116)

where the new operator $J$ is used to denote an integral. More generally,

$$
D^\alpha J^\alpha f = f
$$

(A.117)

$$
J^\alpha D^\alpha f = f + C(x)
$$

(A.118)
where $J^\alpha$ is an integral of order $\alpha$, defined as $J^\alpha = D^{-\alpha}$. Alternatively,

$$D^\alpha D^{-\alpha} f = f \quad (A.119)$$
$$D^{-\alpha} D^\alpha f = f + C(x) \quad (A.120)$$
$$J^\alpha J^{-\alpha} f = f + C(x) \quad (A.121)$$
$$J^{-\alpha} J^\alpha f = f \quad (A.122)$$

**Numerical implementation**

Now on the discrete level, the derivative of any order (including fractional order) can be approximated using finite differences. The finite difference formula based on the forward differencing scheme for a derivative of order $\alpha$ for $\alpha > 0$ can be derived from

$$D_h^\alpha f = \lim_{h \to 0} \left( \frac{1}{h^\alpha} \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^k E_h^{-k} f \right) \quad (A.123)$$

Similarly,

$$D_h^{-\alpha} f = \lim_{h \to 0} \left( \frac{1}{h^{-\alpha}} \sum_{k=0}^{\infty} \binom{\alpha}{k} (-1)^k E_h^{-k} f \right) \quad (A.124)$$

for the backward difference formulation of the differential operator. Implementation is thus slightly complicated because of the infinite upper limit on the summation. This infinite upper limit means, in a physical sense, that the derivative of a function $f$ at time $t = \tau$ depends on the complete history of the signal, for all $-\infty < t < \tau$. This problem can be simplified by taking $h = \frac{t - a}{n}$ where $a$ and $t$ are the lower and upper limits of differentiation, respectively. This allows us to write $n = \frac{t - a}{h}$, so now as $h \to 0$, $n \to \infty$. Now the derivative can be approximated by a finite difference formula assuming $h$ is small. Alternatively, the use of terminals to convert the infinite sum to a finite sum can be considered to be an assumption that the signal is causal, or that the signal or function $f = 0$ for $t < a$, which
is, effectively, the case when dealing with a discrete data set that surely does not extend to \( t = -\infty \). Now the fractional order finite difference approximation using the forward difference is given as

\[
a^D_\alpha f \approx \frac{1}{h^\alpha} \sum_{k=0}^{n} \left( \begin{array}{c} \alpha \\ k \end{array} \right) (-1)^k E_h^{\alpha-k} f, \quad \text{where} \quad n = \frac{t - a}{h} \quad (A.125)
\]

Similarly,

\[
a^D_\alpha f \approx \frac{1}{h^\alpha} \sum_{k=0}^{n} \left( \begin{array}{c} \alpha \\ k \end{array} \right) (-1)^k E_h^{\alpha-k} f, \quad \text{where} \quad n = \frac{t - a}{h} \quad (A.126)
\]

for the backward difference definition of the differential operator. The backward difference definition above corresponds to the discrete implementation of the Grunwald-Letnikov derivative, which is the easiest to numerically implement. As can be seen above, the formula using the backward differencing scheme has a distinct advantage over the forward differencing scheme in that it does not require data at fractional step sizes (the forward difference based model requires data at fractional time steps through the operator \( E_h^{\alpha-k} \)). The forward difference scheme also has the added problem of requiring data at times both before and after the time step where the derivative or integral is being calculated. While this poses no real problem to calculating the derivative or integral given a discrete data set, it does pose a problem from a practical standpoint in that it causes the derivative to depend not only on the signal’s history, but its future as well.

**Linear equations**

To find \( f(t) \) given \( g \) and

\[
\sum_{i=1}^{j} D_\alpha^i f = g \quad (A.127)
\]

along with \( j \) independent conditions on \( f \). This allows for the traditional eigenvalue problem with solutions in terms of the Mittag-Leffler function.
Derivatives and integrals with limits

When performing the typical integration of a function, limits are necessary to eliminate the constant of integration. Limits, or terminals as we will call them, serve the same function when evaluating integrals and derivatives of non-integer order using the shift and difference operators as well as serving another purpose. The differential operator defined in terms of the forward difference operator $\Delta_h^\alpha$, where $\alpha < 0$, is

$$D^\alpha f = \lim_{h \to 0} \left( \frac{\Delta_h^\alpha f}{h^\alpha} \right)$$  \hspace{1cm} (A.128)

$$= \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E^{-k}_h f$$  \hspace{1cm} (A.129)

which, for finite and fixed $n$, tends to the uninteresting limit $D^\alpha f = 0$ as $h \to 0$. Only if, as seen before, this upper limit is set at infinity will an interesting answer be produced. However, if we take $h = \frac{t-a}{n}$ where $t$ and $a$ are the upper and lower terminals of the integro-differentiation, respectively, then $n \to \infty$ as $h \to 0$. This allows us to write

$$a D_t^\alpha f = \lim_{h \to 0} \left( \frac{\Delta_h^\alpha f}{h^\alpha} \right)$$  \hspace{1cm} (A.130)

$$= \lim_{nh = t-a} \frac{1}{h^\alpha} \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E^{-k}_h f$$  \hspace{1cm} (A.131)

which represents the derivative of order $\beta$ if $\alpha = \beta$ and the $\beta$-fold integral if $\alpha = -\beta$. Following the same steps as above, the derivative of order $\beta$ if $\alpha = \beta$ and the $\beta$-fold integral if $\alpha = -\beta$ using the backward difference definition of the differential operator yields

$$a D_t^\alpha f = \lim_{h \to 0} \left( \frac{\nabla_h^\alpha f}{h^\alpha} \right)$$  \hspace{1cm} (A.132)

$$= \lim_{nh = t-a} \frac{1}{h^\alpha} \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E^k_h f$$  \hspace{1cm} (A.133)
A.5 Integral operators

Indefinite integrals

The integral operator has been discussed in short in the previous section, but for the sake of completeness, a thorough discussion will be included herein.

Definition

The indefinite integral has already been defined in terms of the inverse of the derivative. If \( Y \) is a function that when operated upon by \( D^1 \) results in the new function \( y \), then \( Y \) is the integral, or antiderivative, of \( y \) and is denoted by \( D^{-1} y \), i.e.,

\[
\text{if } DY = y, \text{ then } D^{-1} y = Y + C \quad \text{(A.134)}
\]

The integral can also be represented using the notation \( J^n \), which is read the \( n \)-fold integral, or the integral of order \( n \). Using the definition of the derivative based on the forward differencing scheme,

\[
Jf = D^{-1} f + C \quad \text{(A.135)}
\]

\[
= \lim_{h \to 0} \left( \frac{\Delta_h^{-1} f}{h^{-1}} \right) + C \quad \text{(A.136)}
\]

\[
J^n f = D^{-n} f + C(x) \quad \text{(A.137)}
\]

\[
= \lim_{h \to 0} \left( \frac{\Delta_h^{-n} f}{h^{-n}} \right) + C(x) \quad \text{(A.138)}
\]

Using the backward differencing scheme,

\[
Jf = D^{-1} f + C \quad \text{(A.139)}
\]

\[
= \lim_{h \to 0} \left( \frac{\nabla_h^{-1} f}{h^{-1}} \right) + C \quad \text{(A.140)}
\]

\[
J^n f = D^{-n} f + C(x) \quad \text{(A.141)}
\]

\[
= \lim_{h \to 0} \left( \frac{\nabla_h^{-n} f}{h^{-n}} \right) + C(x) \quad \text{(A.142)}
\]
In all of the above cases, $C$ and $C(x)$ are the set of functions or vectors that span the null space of the operators $D^{-1}$ and $D^{-n}$, respectively.

Properties

The integral shares many of the same properties of the derivative.

\begin{align}
(J^n + J^m) f &= J^n f + J^m f \quad (A.143) \\
J^n(f + g) &= J^n f + J^n g \quad (A.144) \\
J^n(\alpha f) &= \alpha J^n f \quad (A.145) \\
J^1 &= J \quad (A.146) \\
J^m J^n f &= J^{m+n} f \quad (A.147)
\end{align}

Extension to negative integers

In the previous section covering differential operators, the extension to negative powers was not made. By now, the reasons why are obvious. Now, before extending the definition of an integral operator to negative integer order, we must first relate integrals of integer order to derivative of negative integer order. The integral can be viewed as the extension of the derivative operator to negative powers. Thus extending the integral to a negative integer power is, in effect, the same as a derivative of integer order, which has already been discussed. Mathematically,

\begin{align}
J^n &= D^{-n} f + C(t) \quad (A.148) \\
&= \lim_{h \to 0} \left( \frac{\Delta^{\frac{n}{h}}}{h^{-n}} \right) + C(t) \quad (A.149)
\end{align}

and

\begin{align}
J^n &= D^{-n} f + C(t) \quad (A.150) \\
&= \lim_{h \to 0} \left( \frac{\nabla^{\frac{n}{h}}}{h^{-n}} \right) + C(t) \quad (A.151)
\end{align}
In both cases, \( C(t) \) exists in the null space of \( D^n \).

In similar fashion to before, an integral of order \( s = -n \) is the same as a derivative of order \( n \),

\[
J^{-n} = D^n \tag{A.152}
\]

which has been discussed previously.

**Extension to reals**

Similarly, integration of order \( \alpha \) is related to differentiation of order \( \alpha \) through the following (for \( \alpha \geq 0 \)):

\[
J^\alpha = D^{-\alpha} + C(t) \tag{A.153}
\]

\[
J^{-\alpha} = D^\alpha \tag{A.154}
\]

**Null space of the indefinite integral operator**

Generally, when considering the integral operator \( J^\alpha \), only for values of \( \alpha \geq 0 \) is \( J^\alpha \) considered to be an integral. For \( \alpha < 0 \), \( J^\alpha \) represents the differential operator. A discussion of the differential operator’s null space, inverse, and implementation has already been included, so we will focus here on \( J^\alpha \) with \( \alpha \geq 0 \).

The null space of the integral operator \( (J^\alpha, \; \alpha \geq 0) \) is defined as \( \mathcal{N}\{J^\alpha\} = \{ f \in F : J^\alpha f = 0 \} \). While not proven here, \( \mathcal{N}\{J^\alpha\} = 0 \), meaning the equation \( J^\alpha f = 0 \) has only the trivial solution \( f = 0 \), and thus the indefinite integral operator has a trivial null space.

**Inverse operators**

Integration is defined as the inverse to a differential operator. As would be expected, the differential operator is the inverse to the integral operator \( J^\alpha \). How-
ever, unlike before, the inverse to the integral operator is unique, which is also expected, as the null space of the integral operator is trivial.

\[
D^\alpha J^\alpha f = f
\]  

(A.155)

Alternatively, the inverse of the indefinite integral operator can be defined as follows. If \( y \) is a function that when operated upon by \( J^\alpha \ (\alpha \geq 0) \) results in the function \( Y + C(x) \), then \( y \) is called the derivative of \( Y + C(x) \) and is denoted by \( D^\alpha \{Y + C(x)\} \) or \( J^{-\alpha} \{Y + C(x)\} \), i.e.,

\[
\text{if } J^\alpha y(x) = Y(x) + C(x), \text{ then } D^\alpha \{Y + C(x)\} = y(x)
\]  

(A.156)

where \( C(x) \in \mathcal{N}\{D^\alpha\} \).

**Numerical implementation**

Implementing the indefinite integral numerically, as defined above, is not an easy task and requires the evaluation of an infinite sum and is still is not unique as any function in the null space of the corresponding differential operator can be added to to the result. For these reasons indefinite integration through discrete formulas is impractical. However, if the definition is changed slightly, implementation of the indefinite integral becomes simple.

**Indefinite integration with terminals**

Indefinite and definite integration are two different operators related by the first fundamental theorem of calculus, which states that if \( f \) is continuous on the interval \([a, b]\) and \( F \) is the antiderivative (or indefinite integral) of \( f \) such that \( F = Jf \), then

\[
\int_a^b f(x)dx = F(a) - F(b)
\]  

(A.157)
From this theorem, it is noteworthy that the indefinite integral operator $J$ operates on a function and returns another function, $J : \mathbb{F} \to \mathbb{F}$, while the definite integral operator, $aJ_b$, where $a$ and $b$ are the upper and lower limits or terminals of the operation, operates on a function and returns a real number, $aJ_b : \mathbb{F} \to \mathbb{R}$.

The second fundamental theorem of calculus holds for a function $f$, continuous on an open interval $I$, and $a$ is any point in $I$, and states that if $F$ is defined by

$$F(x) = \int_a^x f(t)dt$$

then

$$F'(x) = f(x)$$

at each point in $I$. This integral operator with an indefinite terminal, another form of indefinite integration, operates on a function and returns another function, $aJ : \mathbb{F} \to \mathbb{F}$. Using this form of indefinite integration allows us to numerically compute the indefinite integral using a data series.

**Definition**

Denoting the integral operator with an indefinite terminal as $aJ$, the operator is defined by the following. If $Y$ is a function that when operated on by $D^1$ results in the new function $y$, then $Y$ is the integral, or antiderivative of $y$ and is denoted by $aJy$ or $D^{-1}$, i.e.,

$$\text{if } DY = y, \text{ then } aJy = Y$$

(A.160)

Using this definition, the lower terminal $a$ serves as an initial, or boundary condition to eliminate the unknown $C(x)$ that was present before. In this way, $aJ$ is
defined as the inverse to the operator $D$,

\[
aJ f = D^{-1} f
\]  
(A.161)

\[
= \lim_{h \to 0} \left( \frac{\Delta_h^{-1} f}{h^{-1}} \right) 
\]  
(A.162)

but unlike $J$, this operation is unique.

The definition can be extended to positive integers as

\[
aJ^n = D^{-n}
\]  
(A.163)

\[
= \lim_{h \to 0} \left( \frac{\Delta_h^{-n} f}{h^{-n}} \right) 
\]  
(A.164)

Properties

Extension to negative integers

The operator $aJ$ can be extended to negative numbers in the same fashion as $J$. Using the definition as the inverse to the derivative,

\[
aJ^{-n} = aD^n 
\]  
(A.165)

\[
= \lim_{h \to 0} \left( \frac{\Delta_h^{-n} f}{h^{-n}} \right) 
\]  
(A.166)

which is, obviously, just the derivative operator of order $n$.

Extension to reals

Likewise, the extension to all real numbers is straight forward. For $\alpha \geq 0$,

\[
aJ^\alpha = D^{-\alpha}
\]  
(A.167)

\[
= \lim_{h \to 0} \left( \frac{\Delta_h^{-\alpha} f}{h^{-\alpha}} \right)
\]  
(A.168)

\[
aJ^{-\alpha} = D^\alpha
\]  
(A.169)

\[
= \lim_{h \to 0} \left( \frac{\Delta_h^{\alpha} f}{h^{\alpha}} \right)
\]  
(A.170)
Null space of the integral operator with terminals

Like the indefinite integral form discussed previously, the null space of the integral operator with terminals is also trivial. That is $\mathcal{N}\{aJ\} = 0$, or, the only solution to the equation $aJf = 0$ is the trivial solution $f = 0$.

Inverse operators

Having been defined as the inverse to the derivative, it is, once again, not surprising to find the the inverse of the indefinite integral operator with terminals of order $\alpha$ is the differential operator of order $\alpha$. The inverse of $J^\alpha$ is unique and is given by $D^\alpha$. Mathematically,

$$D^\alpha aJ^\alpha f = f$$  \hspace{1cm} (A.171)

Also interesting to note is the fact that

$$aJ^\alpha D^\alpha f = f$$  \hspace{1cm} (A.172)

which is not the same as before. This is another way of saying that $aJ^\alpha$ is the unique inverse of the operator $D^\alpha$.

Numerical implementation

Now that the integral operation (defined as the inverse to the derivative) is unique, it can be implemented numerically. Previously, the operator $aJ^\alpha$ was given as

$$aJ^\alpha = D^{-\alpha}$$  \hspace{1cm} (A.173)

$$= \lim_{h \to 0} \left( \frac{\Delta_h^{-\alpha} f}{h^{-\alpha}} \right)$$  \hspace{1cm} (A.174)
or using the backward difference definition,

\[ a J^\alpha = D^{-\alpha} \]

\[ = \lim_{h \to 0} \left( \frac{\nabla_h^{-\alpha} f}{h^{-\alpha}} \right) \quad (A.175) \]

Now writing the difference operators as a sum gives

\[ a J^\alpha f = \lim_{h \to 0} \left( \frac{\Delta_h^{-\alpha} f}{h^{-\alpha}} \right) \]

\[ = \lim_{h \to 0} h^n \sum_{k=0}^{n} \left( \frac{\alpha}{k} \right) (-1)^k E_h^{\alpha-k} f \quad (A.177) \]

and

\[ a J^\alpha f = \lim_{h \to 0} \left( \frac{\nabla_h^{-\alpha} f}{h^{-\alpha}} \right) \]

\[ = \lim_{h \to 0} h^n \sum_{k=0}^{n} \left( \frac{\alpha}{k} \right) (-1)^k E_h^{-\alpha-k} f \quad (A.179) \]

The integral can now be approximated as

\[ a J^\alpha f = \lim_{h \to 0} h^n \Delta_h^\alpha f \]

\[ \approx h^n \sum_{k=0}^{n} \left( \frac{\alpha}{k} \right) (-1)^k E_h^{\alpha-k} f \quad (A.181) \]

and

\[ a J^\alpha f = \lim_{h \to 0} h^n \nabla_h^\alpha f \]

\[ \approx h^n \sum_{k=0}^{n} \left( \frac{\alpha}{k} \right) (-1)^k E_h^{-\alpha-k} f \quad (A.183) \]

where, for both cases, \( n = \frac{t-a}{h} \). As before, the backward differencing result again has an obvious advantage over the forward difference in that data is only required at regular intervals. When implementing this numerically, then \( h \) is the step size, \( n \) is the number of intervals between \( t \) and \( a \) (or \( n - 1 \) is the number of points), and \( a \) is the starting point for the integration.
Definite integrals

Definite integration is exactly as its name implies: definite. A definite integral is calculated or evaluated over a given interval. A definite integral evaluated over the interval \([a, b]\) is written \(\int_a^b\). The definite integral is merely a special case of the indefinite integral discussed previously, only with a definite upper and lower limit. All of the discussion in the previous section all applies here, but with the upper bound (previously \(x\)) replaced with the constant \(b\). Now, as was discussed before, \(\int_a^b : \mathbb{F} \to \mathbb{R}\). One important difference between the definite and indefinite integral is that the null space of the definite integral is not trivial.

A.6 Numerical implementation summary

We have already briefly discussed the shift, forward difference, backward difference, differential, and integral operators. This section is just a brief summary of the results.

Finite difference formulas

These formulas can be used to numerically implement the following operators:

Shift

Given the data \(f_i = f(a + ih)\), implementation of \(E_h^s f_i\) where \(s \in \mathbb{Z}\) is simply

\[
E_h^s f_i = f_{i+s} \tag{A.185}
\]

Given the data \(f_i = f(a + i\frac{h}{q})\), implementation of \(E_h^\alpha f_i\) where \(\alpha = \frac{p}{q}, \in \mathbb{R}\) is simply

\[
E_h^\alpha f_i = f_{i+p} \tag{A.186}
\]
Note that in order to shift in fractional steps of $h$, the data set must be sampled at a frequency $q/h$. Also note that the formula for $E_h^\alpha$ is simply a special case of $E_h^\alpha$, with $q = 1$. Figure A.1 shows the shift operator $E_h^\alpha$ applied to $f_i = f(a + i\frac{h}{q})$ where $f(t) = t^3 - t^2$ for two different pairs of $\alpha$ and $h$.

**Forward difference**

Given the data $f_i = f(a + i\frac{h}{q})$, implementation of the operator $a \Delta_h^\alpha$ where $\alpha = \frac{p}{q}$, is given by

$$a \Delta_h^\alpha f_i = \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E_h^{-k}f_i$$  \hspace{2cm} (A.187)

$$= \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k f_{i+p-kq}$$  \hspace{2cm} (A.188)

where $n = \frac{t-a}{h}$. Figure A.2 shows a function $f(t)$ and its fractional forward difference or order $\alpha$, calculated using the above formula for two values of $\alpha$.

**Backward difference**

Given the data $f_i = f(a + ih)$, $a \nabla_h^\alpha f_i$ is given by

$$\nabla_h^\alpha f_i = \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E_h^{-k}f_i$$  \hspace{2cm} (A.189)

$$= \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k f_{i-k}$$  \hspace{2cm} (A.190)

where $n = \frac{t-a}{h}$. Figure A.3 shows a function $f(t)$ and its fractional backward difference or order $\alpha$, calculated using the above formula for two values of $\alpha$. 
Figure A.1: The function $f(t) = t^3 - t^2$, the function $f(t + \alpha h)$, and $E_h^\alpha F_i$. Shown for both $\alpha = 0.5$ and $\alpha = 4/3$. 
Differential

Given the data \( f_i = f(a + ih) \), the differential operator of order \( \alpha \) can be implemented as

\[
a D^\alpha f_i \approx \frac{1}{h^{\alpha}} \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E^{-k} h f_i \approx \frac{1}{h^{\alpha}} \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k f_{i-k}
\]  

(A.191)  

(A.192)

where \( n = \frac{t-a}{h} \) is the number of intervals of step size \( h \) between \( t \) and \( a \), or \( n + 1 \) is the number of data points between \( a \) and \( t \).

Figure A.4 shows a function \( f(t) \) with both its analytical derivative of order \( \alpha \) and its derivative of order \( \alpha \) calculated using the above formula, for two different values of \( \alpha \).

Integral

Given the data \( f_i = f(a + ih) \), the integral operator of order \( \alpha \) can be implemented as

\[
a J^\alpha f_i \approx h^{\alpha} \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k E^{-k} h f_i \approx h^{\alpha} \sum_{k=0}^{n} \binom{\alpha}{k} (-1)^k f_{i-k}
\]  

(A.193)  

(A.194)

where \( n = \frac{t-a}{h} \) is the number of intervals of step size \( h \) between \( t \) and \( a \), or \( n + 1 \) is the number of data points between \( a \) and \( t \). Note that the discrete formula for the \( \alpha \)-order integral is the same as the \( \alpha \)-order derivative of order \( -\alpha \), thus the same numerical formula is implemented.

Figure A.5 shows a function \( f(t) \) with both its analytical integral of order \( \alpha \) and its integral of order \( \alpha \) calculated using the above formula, for two different values of \( \alpha \).
A.7 The binomial theorem

In the following $\alpha \in \mathbb{R}; n, k \in \mathbb{N} = \{1, 2, 3, \ldots\}.$

In mathematics, the binomial theorem is an important formula giving the expansion of powers of sums. Its simplest version reads

$$ (A + B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k} \quad (A.195) $$

where $\binom{n}{k}$ are the binomial coefficients defined in the following section. When generalized for any real exponent, the result is

$$ (A + B)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} A^k B^{\alpha-k} \quad (A.196) $$

The preceding formula is known as the generalized binomial theorem and is not only valid for $\alpha \in \mathbb{R},$ but for $\alpha \in \mathbb{C}$ as well.

A.8 Binomial coefficients

In the following $\alpha \in \mathbb{R}; n, k \in \mathbb{N} = \{1, 2, 3, \ldots\}; s \in \mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$

The symbol $\binom{n}{k}$ denotes a binomial coefficient and is commonly read 'n choose k.' While the binomial coefficient is most commonly defined in terms of the factorial, which limits its usefulness to positive integers, its definition can be expanded to allow for both positive and negative non-integer arguments.

Properties

The binomial coefficient possesses the following properties:

$$ \binom{n}{0} = \binom{n}{n} = 1 \quad (A.197) $$
$$ \binom{n}{k} = \binom{n}{n-k} \quad (A.198) $$
\[
\binom{n}{k} = (-1)^k \binom{k-n-1}{k} \quad (A.199)
\]

\[
\binom{n}{k} = \binom{n}{k} \frac{n-k}{k+1} \quad (A.200)
\]

\[
\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} \quad (A.201)
\]

**Positive integers**

For \(n, k \in \mathbb{N}\),

\[
\binom{n}{k} = \frac{n!}{(n-k)!k!} \quad (A.202)
\]

**Positive reals**

Writing the factorial as a gamma function \(\Gamma(n+1) = n!\) allows the binomial coefficient to be generalized to non-integer arguments. Assuming \(\alpha > 0, k \in \mathbb{N}\),

\[
\binom{\alpha}{k} = \frac{\Gamma(\alpha+1)}{k!\Gamma(\alpha-k+1)} \quad (A.203)
\]

\[
\binom{\alpha}{k} = \frac{\alpha(\alpha-1)(\alpha-2)\ldots(\alpha-k+1)}{k!} \quad (A.204)
\]

**Negative integers**

While the Gamma function and factorial are not defined for negative integers, it is still possible to define the binomial coefficient. For \(s < 0\), or \(s = -n\), and \(k \in \mathbb{N}\) the following is true:

\[
\binom{-n}{k} = (-1)^k \frac{(n+k-1)!}{k!(n-1)!} \quad (A.205)
\]

\[
\binom{s}{k} = (-1)^k \frac{(k-s-1)!}{k!(-s-1)!} \quad (A.206)
\]

\[
\binom{s}{k} = \frac{s(s-1)(s-2)\ldots(s-k+1)}{k!} \quad (A.207)
\]
Negative reals

For \( \alpha \in \mathbb{R}, \alpha > 0, \) and \( k \in \mathbb{N} \)

\[
\binom{-\alpha}{k} = (-1)^k \frac{\Gamma(\alpha + k)}{\Gamma(\alpha)k!}
\]

(A.208)

\[
\binom{-\alpha}{k} = -\frac{\alpha(-\alpha - 1)(-\alpha - 2)\ldots(-\alpha - k + 1)}{k!}
\]

(A.209)

All real numbers

For any \( \alpha \in \mathbb{R} \) and \( k \in \mathbb{N} \) the binomial coefficient can be generalized to

\[
\binom{\alpha}{k} = \frac{\alpha(\alpha - 1)(\alpha - 2)\ldots(\alpha - k + 1)}{k!}
\]

(A.210)

\[
= \prod_{n=1}^{k} \frac{\alpha - k + n}{n}
\]

(A.211)

This final expression is known as the generalized binomial coefficient.

A.9 Other forms of the fractional-order integral and derivative

We have already discussed the Grunwald-Letnikov fractional-order derivative and its relationship with the differential, difference, and shift operators. In addition to this form of the fractional-order derivative and integral, one can develop other forms of the operator which have advantages for non-numerical based analysis.

Perhaps the simplest to understand was is the Riemann-Liouville fraction-order integral. It is simply a straightforward extension of Cauchy’s formula for repeated integration,

\[
J^n = \int_a^x \int_a^{\sigma_1} \ldots \int_a^{\sigma_{n-1}} f(\sigma_n) \, d\sigma_n \ldots d\sigma_1 = \frac{1}{(n-1)!} \int_a^x (x - y)^{n-1} f(y) \, dy.
\]

(A.212)

189
As has been previously demonstrated in the development of the Grunwald-Letnikov derivative, by generalizing $n$ to $\alpha$ and the factorial function to the gamma function, this expression can too be generalized to non-integers,

$$J^\alpha = \frac{1}{\Gamma(\alpha)} \int_a^x \frac{f(y)}{(x-y)^{1-\alpha}} \, dy.$$  \hspace{2cm} (A.213)

This is the definition of the Riemann-Liouville integral,

$$J_{RL}^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_a^x \frac{f(t)}{(x-t)^{1-\alpha}} \, dt,$$

which gives the $\alpha$-order integral of the function $f(x)$ analytically. Similarly, an analytical fractional-order derivative can be defined by taking the integer-order derivative of the fractional-order integral. For example, to determine the $1/2$–order derivative of a function, one can simply take the $1/2$-order integral of the function and then apply the differential operator once,

$$D^{1/2} = D^1 J^{1/2}.$$  \hspace{2cm} (A.214)

This form of the fractional-order derivative is known as the Riemann-Liouville derivative and can be written,

$$D_{RL}^\alpha f(x) = \frac{1}{\Gamma(n - \alpha)} \frac{d^n}{dx^n} \int_a^x \frac{f(t)}{(x-t)^{1+\alpha-n}} \, dt, \quad n - 1 \leq \alpha < n.$$

One may of course see a potential problem in this definition. First of all, the fractional-order derivative of a constant is not 0. Secondly, in the solution of a fractional-order differential equation, particularly through the use of Laplace transforms, one might notice that this definition would require the use of fractional-order initial conditions (at the $1/2$-order integral). A solution to this problem was proposed by Caputo in which the derivative is taken first and then fractional-order integration is used to yield a fractional-order derivative. For example, to
determine the $1/2$–order derivative of the function $f(x)$, Caputo would suggest one first apply the integer-order differential operator and then take the fractional-order integral of the result so that

$$D^{1/2} = J^{1/2}D^1.$$  

(A.215)

Now using this definition, initial conditions are only necessary at an integer-order derivative of the function being differentiated.

The Caputo form of the fractional-order derivative can be written as

$$D_C^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{f^{(n)}(t)}{(x-t)^{1+\alpha-n}} dt, \quad n-1 \leq \alpha < n$$

Finally, for the sake of completeness, we will again include the definition of the Grünwald-Letnikov derivative,

$$D_G^\alpha f(x) = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{m=0}^{(x-a)/h} (-1)^m \frac{\Gamma(\alpha + 1)}{m! \Gamma(\alpha - m + 1)} f(x - mh).$$

These are the most commonly encountered definitions for fractional-order integrals and derivatives. Oftentimes when one first hears the term “fractional-order derivative” the natural questions is, “what does it mean?” Oftentimes, people are expecting an answer that relates to the what we are familiar with - position, velocity, and acceleration. But really, those terms are merely names given to specific derivatives or integrals of integer-order, which are defined mathematically in the exact same way as is the fractional-order integral and derivative. In this way, position, velocity, and acceleration are merely names we have given to specific cases of the fractional-order integral and derivative.

One visual example that can be helpful is given in Figure A.6, given by Schumer et al. (2001). In this figure, one can more easily see the continuum of derivatives that exist for a function and how they relate to one another.
Figure A.2: The function $f(t) = t^{2.1}$, its forward difference of order $\alpha$, and the forward difference of order $\alpha$ calculated numerically. Shown for both $\alpha = 1/3$ and $\alpha = -1/2$. 

(a) $\alpha = 1/3$

(b) $\alpha = -1/2$
Figure A.3: The function $f(t) = t^{2.1}$, its backward difference of order $\alpha$, and the backward difference of order $\alpha$ calculated numerically. Shown for both $\alpha = 0.5$ and $\alpha = -2/3$. 
Figure A.4: The function $f(t) = t^{2.1}$, its analytical derivative of order $\alpha$, and the derivative of order $\alpha$ calculated numerically. Shown for both $\alpha = 0.5$ and $\alpha = 4/3$. 

(a) $\alpha = 0.5$

(b) $\alpha = 4/3$
Figure A.5: The function $f(t) = t^{2.1}$, its analytical integral of order $\alpha$, and the integral of order $\alpha$ calculated numerically. Shown for both $\alpha = 0.5$ and $\alpha = 4/3$. 
Figure A.6: The function $f(x) = x^2$ and its fractional-order derivatives ($D_{RL}^\alpha$). Shown in (a), $\alpha = 0.2, 0.4, 0.6, 0.8, 1.0$, and in (b), $\alpha = 1, 1.2, 1.4, 1.6, 1.8, 2.0$. 
BIBLIOGRAPHY


