Comprehensive Simulation of One-Dimensional Unsteady Pipe Network Hydraulics: Improved Formulations and Adaptive Hybrid Modeling

by

Johnathan D. Nault

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy

Department of Civil Engineering
University of Toronto

© Copyright by Johnathan D. Nault, 2017
Comprehensive Simulation of One-Dimensional Unsteady Pipe Network Hydraulics: Improved Formulations and Adaptive Hybrid Modeling

Johnathan D. Nault
Doctor of Philosophy
Department of Civil Engineering | University of Toronto
2017

Abstract

From municipal water supply to wastewater collection, pressurized pipe networks form the basis of many urban water systems. Numerical models are essential for analyzing their unsteady hydraulics, but there are multiple types of models with poorly understood ranges of applicability. Water hammer models feature greater physical accuracy, making them suitable for highly dynamic conditions. They are, however, more computationally demanding than the simpler yet less accurate rigid water column (RWC) and quasi-steady models. Unsteady hydraulic analyses require solution methods that are both physically accurate and computationally efficient; accordingly, there is an accuracy-efficiency trade-off. To balance these competing demands, this thesis presents an adaptive hybrid transient model (AHTM) capable of simulating the full range one-dimensional unsteady pipe network hydraulics.

In developing the comprehensive AHTM, a number of gaps in the literature are addressed. A novel RWC formulation is first proposed, one that overcomes the numerical challenges of previous published work. This is subsequently combined with unsteady flow characterization indices and an adaptive scheme to form an incompressible flow AHTM. Essentially, a more demanding yet more physically accurate model is used only when necessary for efficiency; though powerful, the framework does not consider unsteady-compressible flow. The third objective concerns models for such conditions. A flexible water hammer formulation is introduced that generalizes the method of characteristics and the wave characteristics method, two predominant numerical approaches. Finally, the culmination of this thesis combines each of the aforementioned into a single comprehensive, flexible, and efficient AHTM. The AHTM is shown to adapt to the degree of unsteadiness and, more importantly, individual analyses.

Results of this work ultimately benefit practical analyses of unsteady pipe network hydraulics. Not only can simulations be performed more efficiently, but they may encompass multiple transient flow regimes. Moreover, the key advantages of the comprehensive AHTM are its generality and adaptability.
Acknowledgements

Many parties contributed to this thesis and the development of its contents. Without their support, this work would not be what it is now, so I wish to extend my sincerest gratitude. I thank FP&P HydraTek and the Natural Sciences and Engineering Research Council (NSERC) of Canada for their financial support through my NSERC IPS 1 and IPS 2 scholarships. Of FP&P HydraTek, thanks are due to Fabian Papa for acting as the industrial supervisor for the NSERC scholarships and for his insightful comments, and Djordje Radulj for providing a much needed sense for practicality and engaging talks. Although it goes without saying, a great deal of gratitude is owed to my Ph.D. supervisor, Professor Bryan Karney, for his inspiration, ongoing motivation, and support for both academic and professional endeavors. Lastly, I thank my fiancé Jie Li for her love, continued support, and strength to persevere. Were it not for her, I would not have arrived where I am now.
# Table of Contents

Abstract ................................................................................................................................. ii  
Acknowledgements ............................................................................................................... iii  
List of Tables ........................................................................................................................... viii  
List of Figures ......................................................................................................................... ix  
List of Acronyms .................................................................................................................... xi  

## Chapter 1: Introduction ................................................................................................... 1  
1.1 A Brief Overview of Unsteady Flow Modeling ............................................................... 2  
1.2 Dissertation Significance ............................................................................................... 5  
1.3 Thesis Organization ....................................................................................................... 6  
1.4 Meta-Model Comparative Analysis ............................................................................... 9  
1.5 Related Publications .................................................................................................... 10  

## Chapter 2: Transient Flow Modeling ............................................................................... 11  
2.1 Governing Equations ..................................................................................................... 11  
2.2 Transient Flow Regimes and Conventional Modeling .................................................... 13  
  2.2.1 Unsteady-Compressible Flow .................................................................................. 13  
  2.2.2 Unsteady-Incompressible Flow .............................................................................. 14  
  2.2.3 Quasi-Steady and Steady-State Flow ..................................................................... 16  
  2.2.4 Flow Classification ................................................................................................. 16  
2.3 Pipe Network Modeling Developments ......................................................................... 18  
  2.3.1 Water Hammer Models ......................................................................................... 18  
  2.3.2 Unsteady-Incompressible Flow Models .................................................................. 19  
  2.3.3 Quasi-Steady and Steady-State Flow Models ......................................................... 21  
2.4 Flow Regime Delineation and Adaptive Modeling ......................................................... 23  
2.5 Example – Valve Closure in a Simple System ................................................................. 25  
2.6 Summary ....................................................................................................................... 28  

## Chapter 3: Pipe Network Representation ....................................................................... 29  
3.1 Fundamentals of Graph Theory ..................................................................................... 29  
  3.1.1 Directed Graphs and Incidence Matrices ............................................................... 29  
  3.1.2 State Variables ...................................................................................................... 31  
  3.1.3 Network Representation ......................................................................................... 32  
  3.1.4 State Variables Representation .............................................................................. 33  
  3.1.5 Initial Flow State .................................................................................................... 34  
  3.1.6 Flow Regime De-creation and Adaptive Modeling ............................................... 36  
  3.1.7 Summary ................................................................................................................ 38  

---
3.1.3 Graph Partitioning .................................................................32
3.1.4 A Brief Summary .................................................................35
3.2 Topological Relationships – Null Basis..............................................35
3.3 Application to Pipe Networks ..........................................................40
  3.3.1 Network Properties .............................................................40
  3.3.2 Network Topology ...............................................................41
  3.3.3 Boundary Conditions..............................................................43
3.4 Network Equations ........................................................................46
  3.4.1 Conservation of Mass ..............................................................46
  3.4.2 Conservation of Momentum .....................................................47
3.5 Recent Developments ......................................................................50
3.6 Concluding Remarks .......................................................................52
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow........53
  4.1 Abstract ......................................................................................53
  4.2 Introduction .................................................................................53
  4.3 Background – Unsteady Flow Modeling ..........................................55
    4.3.1 Recent Developments in the Literature .....................................55
    4.3.2 Governing Equations and the Transient Flow Regimes .................56
  4.4 Incompressible Modeling Formulations ...........................................58
    4.4.1 Network Representation and Boundary Conditions .......................58
    4.4.2 Generalized Global Gradient Algorithm .....................................59
    4.4.3 Rigid Water Column Global Gradient Algorithm ..........................61
    4.4.4 Jacobian Submatrix Terms ......................................................62
    4.4.5 Some Remarks on the G-GGA and RWC GGA .............................64
    4.4.6 Implicit Incidence Method .......................................................65
  4.5 Performance of the RWC Formulations ..........................................66
    4.5.1 Example 1 – Small Pipe Network ............................................67
    4.5.2 Example 2 – Skeletonized Water Distribution System ....................71
    4.5.3 Efficiency and Stability ............................................................73
  4.6 Discussion ....................................................................................74
  4.7 Conclusions and Broader Considerations .........................................74
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flow .........................78
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Abstract</td>
</tr>
<tr>
<td>5.2 Introduction</td>
</tr>
<tr>
<td>5.3 Hybrid Global Gradient Algorithm</td>
</tr>
<tr>
<td>5.3.1 Hybrid Incompressible Momentum Equation</td>
</tr>
<tr>
<td>5.3.2 Network Equation and Solution</td>
</tr>
<tr>
<td>5.3.3 Some Remarks on the HGGA</td>
</tr>
<tr>
<td>5.4 Inertial Flow Indicators</td>
</tr>
<tr>
<td>5.4.1 Inertial Energy Indicators</td>
</tr>
<tr>
<td>5.4.2 Inertial Junction Head</td>
</tr>
<tr>
<td>5.4.3 Utility and Use with the HGGA</td>
</tr>
<tr>
<td>5.5 Adaptive Scheme and Implementation</td>
</tr>
<tr>
<td>5.6 Simulation of Incompressible Flow</td>
</tr>
<tr>
<td>5.6.1 Example 1 – Pumping Pipeline</td>
</tr>
<tr>
<td>5.6.2 Example 2 – EPS of a Transmission Network</td>
</tr>
<tr>
<td>5.6.3 Computational Efficiency</td>
</tr>
<tr>
<td>5.7 Discussion and Critical Appraisal</td>
</tr>
<tr>
<td>5.8 Conclusions and Broader Implications</td>
</tr>
<tr>
<td><strong>Chapter 6:</strong> Generalized Characteristic Method for Modeling Water Hammer</td>
</tr>
<tr>
<td>6.1 Abstract</td>
</tr>
<tr>
<td>6.2 Introduction</td>
</tr>
<tr>
<td>6.3 Unsteady Flow Modeling</td>
</tr>
<tr>
<td>6.3.1 Governing Equations and Characteristic Representation</td>
</tr>
<tr>
<td>6.3.2 Quasi-Steady Friction Integral Approximations</td>
</tr>
<tr>
<td>6.3.3 Unsteady Energy Dissipation</td>
</tr>
<tr>
<td>6.4 Generalized Characteristic Method</td>
</tr>
<tr>
<td>6.4.1 Compatibility Equations</td>
</tr>
<tr>
<td>6.4.2 Interior and Junction Analyses</td>
</tr>
<tr>
<td>6.4.3 Variable Reach Scheme</td>
</tr>
<tr>
<td>6.5 Computational Accounting</td>
</tr>
<tr>
<td>6.5.1 Comparison of Numerical Methods</td>
</tr>
<tr>
<td>6.5.2 Accuracy and Efficiency</td>
</tr>
<tr>
<td>6.6 Unified Simulation of Transient Flow</td>
</tr>
</tbody>
</table>
List of Tables

Table 1.1: Comparison of 1-D unsteady flow models’ key characteristics .................................................4
Table 4.1: Dimensions and initial flows for pipe network 1 ...........................................................................67
Table 4.2: Formulation run times relative to those of the G-GGA .................................................................68
Table 4.3: RWC GGA and IIM pipe network 1 run times relative to those of the G-GGA ....................73
Table 5.1: Initial conditions for pipe network 2 ..............................................................................................94
Table 5.2: EPS operations log for pipe network 2 ..........................................................................................95
Table 5.3: HGGA EPS relative run times for pipe network 2 .......................................................................98
Table 6.1: Interior analysis unit FLOPs and minimum $N_R$ for different configurations ..........119
Table 6.2: Junction analysis unit FLOPs for the generalized characteristic method ..................119
Table 6.3: Example 2 interior analysis FLOPs per time step .................................................................126
Table 6.4: Example 3 interior analysis FLOPs per time step .................................................................130
Table 7.1: Comparison of the current AHTM against previous AHTMs ..............................................137
Table 7.2: Simulation parameters for time steps and indicator tolerances ........................................152
Table 7.3: Example 2 number of solution time steps for each model .................................................157
Table 8.1: Comparison of current RWC formulations against previous work ..............................165
Table 8.2: Comparison of current water hammer formulations against previous work ...........166
Table 8.3: Comparison of present adaptive hybrid transient models against others ...............168
List of Figures

Figure 1.1: Model schematic of a water distribution system .................................................. 1
Figure 1.2: Illustration of water hammer effect due to sudden valve closure ......................... 2
Figure 1.3: Conceptual relationship between the different types of models ............................. 3
Figure 1.4: Transient flow regimes and relative importance of dynamic effects ..................... 5
Figure 1.5: Generalization of quasi-steady, RWC, and water hammer models to hybrid models .8
Figure 2.1: Relative importance of flow characteristics ....................................................... 17
Figure 2.2: Example pipe network showing base network and discretized reaches .................. 19
Figure 2.3: Reservoir-valve-reservoir system ................................................................. 26
Figure 2.4: Junction J1 head traces for various valve operations ........................................... 27
Figure 3.1: Undirected and directed graphs (Strang, 1987) .................................................. 30
Figure 3.2: Incomplete and complete sets of loops .............................................................. 31
Figure 3.3: Two spanning trees for graph $G_2$ (Strang, 1987) ............................................ 33
Figure 3.4: A graph and its core, external forest, and bridges ............................................. 34
Figure 3.5: Graph $G_D$ and one of its subgraphs $G_S$ ....................................................... 34
Figure 3.6: Spanning tree and chords for null space example ............................................. 39
Figure 3.7: Sample pipe network schematic and graph (Onizuka, 1986) ............................... 42
Figure 3.8: Sample pipe network graph with pseudo-links and loops (Onizuka, 1986) .......... 42
Figure 3.9: Nodal continuity – net inflow and outflow ....................................................... 46
Figure 4.1: Pipe network 1 (Onizuka, 1986) .................................................................. 67
Figure 4.2: Simulation results for pipe network 1 with a tank diameter of 10 m .................. 69
Figure 4.3: Simulation results for pipe network 1 with a tank diameter of 1 m .................... 70
Figure 4.4: Pipe network 2 – water distribution system ....................................................... 71
Figure 4.5: Simulation results for pipe network 2 .............................................................. 72
Figure 5.1: HGGA variable time-stepping scheme ............................................................. 86
Figure 5.2: Adaptive hybrid incompressible flow model implementation ............................. 87
Figure 5.3: Topology and characteristics of pipe network 1 .............................................. 88
Figure 5.4: Simulation results for pipe network 1 ............................................................... 89
Figure 5.5: Inertial energy characteristics for pipe network 1 ............................................. 90
Figure 5.6: Head components for the pump discharge junction, pipe network 1 ................. 91
Figure 5.7: HGGA simulation results for pipe network 1 ................................................... 92
Figure 5.8: Pipe network 2 – water transmission network ................................................... 93
Figure 5.9: Demand pattern for pipe network 2 ................................................................. 94
Figure 5.10: Pipe network 2 EPS results – Zone 1 pump station 1 discharge head .......... 96
Figure 5.11: Pipe network 2 EPS results – Zone 1 pump station 2 discharge head .......... 96
Figure 5.12: Pipe network 2 EPS results – Zone 1 pump station 4 discharge head .......... 97
Figure 5.13: Pipe network 2 EPS results – Zone 1 elevated tower head ......................... 97
Figure 6.1: Characteristic representation of wave space-time paths ................................ 107
Figure 6.2: Finite difference approximations for the MIAB unsteady friction model ....... 111
Figure 6.3: Characteristic lines for hydraulic solutions at interior and end nodes .......... 113
Figure 6.4: Staggered variable reach computational grids for a pipe with $N_0 = 4$ ........ 114
Figure 6.5: Example 1 – idealized reservoir-pipeline-valve system .............................. 115
Figure 6.6: Comparison of fixed space-time computational grids without interpolation 117
Figure 6.7: Comparison of friction representations for a pipe modeled with $NR = 2$ ........ 118
Figure 6.8: Relative head extrema error for example 1 using the generalized solution .... 122
Figure 6.9: Comparison of simulation results for example 1 with $R = 0.25$ ............... 123
Figure 6.10: Example 2 pipe network (adapted from Streeter and Wylie, 1967) .......... 124
Figure 6.11: Example 2 results for the valve head following sudden valve closure ..... 125
Figure 6.12: Example 3 water distribution system showing transmission network in red 127
Figure 6.13: Example 3 simulation results with unsteady friction and $\Delta t = 0.02 \text{ s}$ .... 128
Figure 6.14: Example 3 simulation results with steady friction and $\Delta t = 0.02 \text{ s}$ ..... 129
Figure 7.1: Physical accuracy and computational efficiency for pipe network models ....... 135
Figure 7.2: Hybrid compressible-incompressible flow representation of an arbitrary pipe $j$ ... 140
Figure 7.3: Pipe network 1 showing element types (adapted from Onizuka, 1986) .......... 143
Figure 7.4: Transient flow regimes and relative importance of dynamic effects .......... 147
Figure 7.5: Event-oriented adaptive implementation of the CGGA ............................... 149
Figure 7.6: Comprehensive adaptive hybrid transient model implementation ............. 151
Figure 7.7: Example 1 simulation results for node J2 ....................................................... 153
Figure 7.8: Pipe network 2 – water transmission network (Nault and Karney, 2016b) .... 154
Figure 7.9: Example 2 EPS results for Zone 1 Pump Station 1 discharge head .......... 155
Figure 7.10: Example 2 EPS results for Zone 1 Pump Station 2 discharge head .......... 155
Figure 7.11: Example 2 EPS results for Zone 1 Pump Station 4 discharge head .......... 156
Figure 7.12: Example 2 EPS results for Zone 1 Pump Station 5 discharge head .......... 156
Figure 7.13: Example 2 power failure results for Zone 1 Pump Station 1 discharge head ..... 158
Figure 7.14: Example 2 power failure results for Zone 1 Pump Station 2 discharge head ..... 158
Figure 7.15: Example 2 power failure results for Zone 1 Pump Station 4 discharge head .......159
Figure 7.16: Example 2 power failure results for Zone 1 Pump Station 5 discharge head .......159
Figure 8.1: Conceptual relationship between the different types of models ..........................164
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHTM</td>
<td>Adaptive hybrid transient model</td>
</tr>
<tr>
<td>AMG</td>
<td>Algebraic multigrid (method)</td>
</tr>
<tr>
<td>AWH</td>
<td>Algebraic water hammer</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational fluid dynamics</td>
</tr>
<tr>
<td>CFL</td>
<td>Courant-Freiderichs-Lewy (stability criterion)</td>
</tr>
<tr>
<td>CG</td>
<td>Conjugate gradient (method)</td>
</tr>
<tr>
<td>CGGA</td>
<td>Comprehensive global gradient algorithm</td>
</tr>
<tr>
<td>EIM</td>
<td>Explicit incidence method</td>
</tr>
<tr>
<td>EPS</td>
<td>Extended period simulation</td>
</tr>
<tr>
<td>G-GGA</td>
<td>Generalized global gradient algorithm</td>
</tr>
<tr>
<td>GAWH</td>
<td>Global algebraic water hammer</td>
</tr>
<tr>
<td>GCM</td>
<td>Generalized characteristics method</td>
</tr>
<tr>
<td>GGA</td>
<td>Global gradient algorithm</td>
</tr>
<tr>
<td>HGGA</td>
<td>Hybrid global gradient algorithm</td>
</tr>
<tr>
<td>IIM</td>
<td>Implicit incidence method</td>
</tr>
<tr>
<td>MIAB</td>
<td>Modified instantaneous acceleration-based (unsteady friction model)</td>
</tr>
<tr>
<td>MOC</td>
<td>Method of characteristics</td>
</tr>
<tr>
<td>ODE</td>
<td>Ordinary differential equation</td>
</tr>
<tr>
<td>PCG</td>
<td>Preconditioned conjugate gradient (method)</td>
</tr>
<tr>
<td>PDE</td>
<td>Partial differential equation</td>
</tr>
<tr>
<td>RWC</td>
<td>Rigid water column</td>
</tr>
<tr>
<td>WCM</td>
<td>Wave characteristics method</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

Pipe networks form the basis of many urban water infrastructure systems. Municipal water distribution networks deliver potable water to end users through arrays of interconnected pipes, and wastewater piped collection systems convey sanitary and storm flows away from the urban environment. Indeed, such systems are crucial utilities. Pipe networks are rather complex: they span large distances and often comprise numerous elements (e.g., Figure 1.1). Consequently, their hydraulic design, operation, and analysis are equally complex.

Figure 1.1: Model schematic of a water distribution system

Depending on the conditions, pipe network hydraulics may be steady or unsteady. The latter are a transitional or dynamic domain, whereby the hydraulics vary with time until dynamic effects dissipate. Unsteady or transient conditions arise due to changes at the boundaries of a pipe network (e.g., closing a valve to stop flow). This poses a particular challenge for pressurized systems. The unsteadiness resulting from boundary changes may be mild, or it may be abrupt with potentially detrimental consequences. Because water is dense and only slightly compressible, even small changes in flow can produce significant changes in head. One such phenomenon is water hammer (Figure 1.2), where sudden flow stoppage produces a pressure wave that propagates
away from its source. If not properly managed, the associated extreme pressures can damage infrastructure, lead to significant financial loss, and negatively impact water quality; accordingly, it is imperative to adequately consider unsteady flow conditions.

Figure 1.2: Illustration of water hammer effect due to sudden valve closure

Fundamental to analyzing pressurized pipe network hydraulics are one-dimensional unsteady flow models. These simulate unsteady flow conditions to assess risk and design protective measures. Although insightful, multiple types of transient flow models exist, and each is suited to different degrees of unsteadiness. Those of greater physical accuracy are more computationally demanding, making them less applicable than their simpler but less accurate counterparts; and yet, greater physical accuracy is sometimes necessary. Thus, there is a trade-off between physical accuracy and computational efficiency. To balance these competing demands, this thesis presents a comprehensive modeling formulation capable of accurately and efficiently simulating the full range of one-dimensional unsteady pressurized pipe network hydraulics.

1.1 A Brief Overview of Unsteady Flow Modeling

Models of fluid flow vary widely in their physical basis. Some are capable of simulating one-, two-, and three-dimensional fluid flow; compressible and incompressible flow; single- and multi-phase fluids; and steady and unsteady conditions. Applications also range in physical scale, from the very small (microfluidic behaviour for biomedical applications) to the very large
(atmospheric, oceanic, and climatic behaviour). Pipe networks fall in the middle. These systems may have free-surface, pressurized, or mixed flow conditions; the subject of this thesis is pressurized pipe networks, such as modern water supply systems. Throughout the remainder of this thesis, “pipe network” refers to those pressurized systems.

Analyses of pressurized pipe network hydraulics are simplified by a few key assumptions. Perhaps the most important are that the fluid is single-phase and the flow dynamics are one-dimensional, but even with these, simulation remains challenging. In addition to having intricate topologies that span large distances (on the order of $10^2$ to $10^3$ km), pipe networks host a range of complex phenomena, including cavitation, fluid column separation, and unsteady friction. Further, model selection is complicated because multiple types of models exist (Figure 1.3): each features a different range of physical validity due to differing assumptions of the flow physics.

![Figure 1.3: Conceptual relationship between the different types of models](image)

Unsteady flow models are categorized according to their underlying physical assumptions. These assumptions include those of fluid compressibility and conduit elasticity (collectively referred to as “compressibility effects”), fluid inertia, and viscous fluid resistance (i.e., head losses). Water hammer models are at one extreme of the modeling spectrum – they consider all of the aforementioned, plus unsteady friction effects. Comparatively, quasi-steady (also called non-inertial unsteady and quasi-unsteady) models neglect dynamic (compressibility and inertial) effects. They are physically valid for slow-varying flow conditions. So-called intermediate rigid water column (RWC) models span the middle ground, which consider inertial but not compressibility effects. One-dimensional unsteady flow models clearly are well-defined (Table 1.1). However, their physical accuracy (the degree to which a model adequately represents the flow physics) remains unclear due to the complex nature of unsteady pipe network hydraulics.
Chapter 1: Introduction

<table>
<thead>
<tr>
<th>Model</th>
<th>Physical Basis</th>
<th>Time Step (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Resistance</td>
<td>Inertia</td>
</tr>
<tr>
<td>Quasi-Steady</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>RWC</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Water Hammer</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

Table 1.1: Comparison of 1-D unsteady flow models’ key characteristics

Conventionally, model selection is made according to the application. Long-term analyses concern hydraulic variations over the course of hours to days; quasi-steady models are often used here with time steps on the order of 10² to 10³ s. Applications include capacity evaluation, and operational optimization. In each case, compressibility and inertial effects are rarely of interest, so quasi-steady models are the ideal alternative. Dynamic effects are not necessarily negligible though; rather, they are merely neglected. Dynamic effects are typically simulated using water hammer models when the structural integrity of a system’s elements is concerned. An example analysis is that of extreme pressures resulting from rapid boundary condition changes (“rapid” implies a large magnitude change over a short duration, such as sudden valve closure). Compared to quasi-steady models, water hammer models are significantly more computationally intensive. They use much smaller time steps (often on the order of 10⁻⁴ to 10⁻² s), and they additionally require the spatial discretization of individual pipes, two limitations that restrict their practical application. Even RWC models, with time steps on the order of 10⁻¹ to 10¹ s, feature modest computational demand. Both accuracy and efficiency are crucial to practical modeling.

Indeed, the conventional approach to model selection is sensible but certainly not optimal. Unlike distinct differences between the types of models, dynamic effects are not confined to specific regions. They are, in fact, transitional over a continuous spectrum (Karney, 1990; Figure 1.4). Fluid compressibility and inertia may be important during one period, but they gradually dissipate. In a given application, the conventional approach relies on the most suitable type of model in terms of physical accuracy, that is, one that provides the required level of physical accuracy; to do otherwise introduces the possibility of overlooking potentially critical hydraulics. Though, it is not always practical to use a more physically accurate model when its greater accuracy is only needed momentarily, for there is a trade-off between computational efficiency and physical accuracy. The overarching goal of model selection is to thus choose a type of model that provides a reasonable representation of the underlying dynamics, according to the purpose of the analysis, without incurring significant computational expense beyond what is necessary.
1.2 Dissertation Significance

Practical analyses require models that are both physically accurate and computationally efficient. One approach to balancing these competing demands is adaptive hybrid modeling: adaptive hybrid modeling combines two or more types of models with an adaptive scheme that selects the most appropriate type of model at each step throughout a simulation. In this way, the more computationally demanding types of models are only used when their greater accuracy is warranted. The concept of adaptive hybrid modeling is not novel to this work. To the author’s knowledge, Axworthy (1997) and Filion and Karney (2002) are the first instances of adaptive hybrid transient models (AHTMs; the reader is referred to Chapter 2 for a detailed review of these references). This dissertation develops a similar AHTM, but one that overcomes the limitations of previous efforts, the individual steps of which are novel.

Fundamental to the current AHTM are a comprehensive modeling formulation, unsteady flow characterization, and an adaptive scheme. These facilitate accurate and efficient simulations. Essentially, the AHTM uses more physically accurate yet computationally demanding water hammer model only when dynamic effects are present, whereas slow-varying conditions are represented using a more efficient RWC or quasi-steady solver. Though similar to Axworthy (1997) and Filion and Karney (2002), the current work distinguishes itself by addressing the following limitations of previous studies:

1. Existing RWC formulations are both computationally inefficient and prone to numerical
instabilities. Moreover, there have been no recent efforts to address this, predominantly due to the perception that such models have little utility. The first research objective is to thus improve the functionality of RWC formulations and demonstrate how they are a practical modeling alternative.

2. Numerous studies have sought to characterize unsteady pressurized flows. Such studies consider limited operations, overlook the role of inertial effects, or fail to define firm bounds. Accordingly, novel unsteady flow characterization indices are introduced to not only describe unsteady flows but also guide model selection.

3. Two prevalent water hammer solution methods, the method of characteristics (MOC) and the wave characteristics method (WCM), are often contrasted on their performance. Both have unique advantages: the MOC features greater solution resolution, whereas the WCM is more efficient. To benefit from the strengths of each, this dissertation generalizes the two approaches to a single common formulation capable of being implemented as either or a combination of the MOC and WCM.

4. Though frequently treated in isolation from one and other, this work shows how quasi-steady, RWC, and water hammer models can be formulated using a common mathematical framework, thus enabling the development of an intuitive AHTM.

5. Finally, the last research objective integrates each of the above items. It concerns developing a physically accurate, computationally efficient, comprehensive model for simulating the full range of one-dimensional unsteady pressurized flows.

Central to the above contributions, and thus this dissertation, are improving physical accuracy and computational efficiency. The significance of this is twofold. First, characterizing unsteady flows provides a means of gauging model physical validity, thereby aiding model selection. Secondly, by permitting more efficient simulations, analysts can explore a wider range of scenarios, a crucial factor for analyzing worst-case conditions, design optimization, and model calibration. Moreover, this greatly benefits the analysis of larger and more complex networks. The findings of this work allow more practical unsteady flow modeling.

1.3 Thesis Organization

This thesis is partitioned between the fundamentals (Chapters 2 and 3) and its core contributions (Chapters 4 through 7). Chapter 2 reviews pressurized pipe network hydraulics,
including unsteady flow physics and the literature, and Chapter 3 discusses the role of graph theory in pipe network modeling. Graph theory permits the mathematical analysis of arbitrary pipe networks, forming the basis of much of this work. Chapters 4 through 7 build on Chapters 2 and 3 while addressing the research objectives.

Chapter 4 covers the first research objective. Existing RWC modeling formulations suffer numerical instabilities due to using explicit numerical schemes to integrate a system of stiff differential equations. To improve upon this, two improved RWC formulations are presented for simulating unsteady-incompressible flow: the RWC global gradient algorithm (GGA), and the implicit incidence method (IIM). Both are unique to this work. The RWC GGA represents an extension of the work of Todini (2011) and Giustolisi et al. (2012), while the IIM combines Shimada’s (1989) incidence method with implicit numerical integration to ensure solution stability. Using two pipe networks, one simple and the other moderately complex, the two formulations are shown to have improve stability and efficiency over previous work. Furthermore, the chapter highlights the practical utility of RWC models.

As an extension to Chapter 4, Chapter 5 addresses the accuracy-efficiency tension for incompressible flow modeling. An incompressible flow AHTM is developed that uses the hybrid GGA (HGGA), a novel solver. The HGGA generalizes the RWC GGA and generalized GGA (Giustolisi et al., 2012), so it can be adjusted to simulate hydraulics ranging from steady-state to unsteady-incompressible flow. To balance computational efficiency and physical accuracy, the HGGA is combined with an adaptive scheme that actively adjusts the model and time step according to the simulated conditions. A small time step is used when inertial effects are deemed present, and a large time step is used for quasi-steady flow conditions. Two examples illustrate how the AHTM provides a better representation of unsteady flow using a quasi-steady model. While shown to perform well, even when compared against a water hammer model for slow varying conditions, the incompressible flow AHTM requires greater computational effort than a quasi-steady model. Nevertheless, the incompressible flow AHTM can be modified to suit the simulation needs be them physically- or analysis-based.

Unlike Chapters 4 and 5, Chapter 6 concerns the simulation of unsteady-compressible network flows. Unsteady-compressible flow or water hammer simulations are typically performed using the method of characteristics (MOC; Wylie and Streeter, 1978; Chaudhry, 2014) or the wave characteristics method (Wood et al., 1966; wood, 2005), each of which features a comparable advantage over the other. The MOC yields higher resolution solutions, whereas the
WCM’s simplifications permit more efficient simulations. To benefit from the advantages of both, Chapter 6 presents a novel solution approach, termed the generalized characteristic method (GCM), that generalizes the MOC and WCM. By combining the two, the GCM can implement either or a combination thereof. Three examples show that the GCM yields comparable results to the MOC but at a computational cost closer to (and sometimes even better than) the WCM.

Chapter 7 represents the culmination of this thesis. It presents a comprehensive AHTM for simulating the full range of unsteady hydraulics. This work is analogous to that of Axworthy (1997), but the current AHTM distinguishes itself by combining a novel comprehensive solver with transient flow characterization and an adaptive scheme. The solver, termed the comprehensive global gradient algorithm (CGGA), combines the GCM and the HGGA, much like how the HGGA encompasses the RWC GGA and generalized GGA (Giustolisi et al., 2012). Figure 1.5 illustrate the conceptual relationship between the individual types of models and generalized AHTMs. With the CGGA, individual links are simulated with either compressible or incompressible flow, and arbitrary coupled boundary conditions are solved inherently. An improved transient flow indicator over that of Chapter 5 is presented to characterize the importance of dynamic effects. Together with the CGGA and an adaptive scheme, the comprehensive AHTM adjusts the solution according to the degree of unsteadiness. Altogether, the methodology is shown to be computationally efficient, physically accurate, numerically stable, and, moreover, general and adaptable.

*Figure 1.5: Generalization of quasi-steady, RWC, and water hammer models to hybrid models*
Finally, Chapter 8 summarizes the overall findings and discusses future work. Potential next steps are discussed, including combining the current work with high performance computing techniques (e.g., parallel computing) and advanced concepts from graph theory to further improve computational efficiency; coupling the current work with a water quality model would also prove beneficial for analyses of potable water supply systems. Perhaps the biggest future step is implementing the findings here within an available modeling software, for the overall advantage of the current work is its generality and ability to balance accuracy and efficiency.

1.4 Meta-Model Comparative Analysis

Central to this research are different types of models (e.g., water hammer and quasi-steady models) and different numerical formulations of a given type of model (e.g., two numerical approaches for water hammer models). Throughout this dissertation, meta-model comparative analyses are used to evaluate model performance; essentially, a subject model is validated by contrasting it against a proven and more accurate (physically or numerically) model. Though not as rigorous as the verification and validation frameworks adopted in the broader computational fluid dynamics community (Oberkampf and Trucano, 2002; Roy and Oberkampf, 2011), meta-model comparisons are well-suited here given the applications’ context. That is, pipe network models are applied to systems that vary widely in configuration, hydraulics, and characteristics, and each features an often unknown degree of uncertainty. Thus, pipe network models are more of a decision tool rather than a precise representation of the true system.

In performing meta-mode comparisons, the following definitions are introduced:

- **Computational efficiency**: characterized by run time, computational efficiency represents the inverse of the computational effort required by a model to perform a simulation. Amongst other factors (e.g., computing power), computational efficiency directly relates to the number of operations required by a model to perform a simulation.

- **Numerical accuracy**: numerical accuracy pertains to the adequacy of a model’s solution of its governing mathematical framework. Spatial and temporal discretization, numerical iterations, truncation error, and round off error each affect numerical accuracy.

- **Physical accuracy**: in this work, physical (model) accuracy concerns the validity of a model’s underlying physical assumptions (namely those of dynamic effects). For example, an RWC model is only physically valid when fluid compressibility effects are negligible, otherwise the model yields physically invalid results. In contrast, a water
hammer model has the greatest physical accuracy of models considered in this work.

1.5 Related Publications

During the course of the author’s graduate studies, four peer-reviewed articles were co-authored with the author’s supervisor, Professor Bryan Karney. The author conducted the research and prepared the manuscripts, while Professor Karney provided guidance and critical review. Chapters 4 through 7 are based on these articles. As of the writing of this thesis, the first three articles have been published, one of which as a part of a conference’s proceedings. Of the last two articles, the first has been accepted for publication, while the remaining manuscript corresponding to Chapter 7 was submitted to IAHR’s Journal of Hydraulic Research on 2017-09-11. For reference, citations for the first four publications are provided below:


**Chapter 6**: Nault, J. D.; Karney, B. W.; and Jung, B.-S. (accepted on 2017-09-05 by ASCE J. Hydraul. Eng.). “Generalized flexible solution method for transient pipe network hydraulics.”

Chapter 2: Transient Flow Modeling

As with any problem, it is helpful to begin with a review of the fundamentals. Chapter 2 presents a background on modeling transient flows and recent developments. The governing equations are first introduced, and they are shown to reduce to different forms depending on which physical assumptions are made. This directly relates to the transient flow regimes and thus the types of unsteady flow models. Recent modeling developments are then reviewed – the objective of this is to connect the often isolated realms of steady and unsteady modeling.

2.1 Governing Equations

One-dimensional unsteady pipe network flow models are largely based on the same two governing equations. These expressions encapsulate the underlying physics of one-dimensional unsteady-compressible flow, namely the conservation of linear momentum and mass continuity for slightly compressible fluids such as water. The governing equations of mass and momentum are reviewed below: for detailed derivations, the reader is referred to standard texts (Wylie and Streeter, 1978; Chaudhry, 2014).

Of the two governing equations, that for momentum represents the effects of fluid inertia and viscous fluid resistance. It is obtained by applying Newton’s second law to an elementary control volume, which yields the expression

\[
\frac{1}{g} \frac{\partial V}{\partial t} + \frac{\partial H}{\partial x} + \frac{4}{\rho g D} \tau + \frac{1}{g} \nu \frac{\partial V}{\partial x} = 0
\]  

[2.1]

where \( V = V(x, t) \) is the mean flow velocity along the pipe (m/s), \( H = H(x, t) \) is the head along the pipe (m), \( D \) is the pipe’s inner diameter (m), \( \tau \) is the fluid’s wall shear stress (Pa), \( \rho \) is the fluid’s density (kg/m\(^3\)), \( g \) is the acceleration due to gravity (m/s\(^2\)), \( x \) is the distance along the pipe (m), and \( t \) is time (s). The first term represents the effects of fluid inertia, that is, the tendency to resist acceleration and deceleration. Viscous fluid resistance, which is purely dissipative, is represented by the third term. Unlike the aforementioned, the fourth advective term is seldom considered for reasons that are discussed later. Together, Equation [2.1] relates the fluid inertia and viscous resistance quantities to the spatial variation in head, \( \partial H/\partial x \).

Complementary to Equation [2.1] is the continuity equation. It more directly represents the effects of fluid compressibility, and the continuity expression is obtained by applying the conservation of mass to a control volume within a Lagrangian frame of reference, which leads to
Chapter 2: Transient Flow Modeling

\[ \frac{a^2 V}{g} \frac{\partial V}{\partial x} + \frac{\partial H}{\partial t} + V \frac{\partial H}{\partial x} - V \sin(\alpha) = 0 \]  

[2.2]

where \( a \) = the pipe’s wave speed (m/s), and \( \alpha \) = the pipe’s inclination from the horizontal (rad). Equation [2.2] relates spatial variations in \( V \), temporal changes in \( H \), and more advective terms.

The wave speed \( a \) represents the sonic speed at which disturbances propagate through a pipe. It depends on many factors, most of which are conveniently summarized in the expression below:

\[ a = \sqrt{\frac{1}{\rho} \left( \frac{KE'}{K+E'} \right)} \quad E' = \left( \frac{e}{D} \right) E \]  

[2.3]

where \( K \) = the bulk fluid modulus (GPa), \( E' \) = the wall thickness-weighted elastic modulus of the pipe (GPa), \( E \) = elastic modulus of the pipe material (GPa), and \( e \) = the pipe wall thickness (m).

From Equation [2.3], the wave speed depends on a fluid’s compressibility, the pipe wall elasticity, the dimensionless ratio \( e/D \), and the fluid density. The expression is valid for a structurally unrestrained pipe, though modifications can be made for buried and restrained pipes, as well as those containing entrapped or dissolved air (see Chaudhry, 2014).

Equations [2.1] and [2.2] are expressed in terms of the dependent variables \( H \) and \( V \) and the independent variables \( x \) and \( t \). Given \( V = Q/A \), it is often more convenient to express the governing equations in terms of \( H \) and \( Q \) as

\[ \frac{1}{gA} \frac{\partial Q}{\partial t} + \frac{\partial H}{\partial x} + J_{QF} + J_{UF} + \frac{1}{gA} V \frac{\partial Q}{\partial x} = 0 \quad J_{QF} = \frac{f}{2gDA^2} |Q|Q \]  

[2.4]

and

\[ \frac{a^2 Q}{gA} \frac{\partial Q}{\partial x} + \frac{\partial H}{\partial t} + V \frac{\partial H}{\partial x} - V \sin(\alpha) = 0 \]  

[2.5]

respectively, where \( Q = Q(x, t) \) is the flow rate along the pipe (m\(^3\)/s), \( A \) = the pipe’s cross-sectional area (m\(^2\)), \( J_{QF} \) = the unsteady friction unit head loss, and \( J_{UF} \) = the quasi-steady friction unit head loss. Together, \( J_{UF} \) and \( J_{QF} \) represent viscous energy dissipation. Often it is sufficient to only consider \( J_{QF} \): \( J_{UF} \) is only important under highly unsteady flows where rapid fluctuations in \( V \) give rise to greater energy dissipation. There are a number of unsteady friction models in the literature (e.g., Axworthy et al., 2000; Vítkovský et al., 2006), so \( J_{UF} \) is left unevaluated for now.

Equations [2.4] and [2.5] constitute a set of non-linear hyperbolic partial differential equations (PDEs) that require some rather clever numerical approaches to solve.

Though derivations of the governing equations are omitted here, it is important to emphasize
Chapter 2: Transient Flow Modeling

the following inherent assumptions:

- Flow is one-dimensional axisymmetric;
- Conditions are isothermal (i.e., temperature changes are negligible);
- The domain comprises a single-phase fluid;
- The wave speed is constant;
- The fluid and pipe wall behave as a linear-elastic material, and;
- Flow within a conduit remains fully pressurized.

Whenever applying a model the validity of these assumptions must be considered. For pipe network hydraulics, these assumptions are often reasonable, but there are indeed cases where they do not hold true. When invalid, these assumptions can lead to erroneous model results. In the following section, Equations [2.4] and [2.5] are shown to reduce to a different form for each transient flow regime depending on which additional physically-based assumptions are made.

2.2 Transient Flow Regimes and Conventional Modeling

Unsteady flows are classified by considering transient flow regimes. These represent differing degrees of unsteadiness, and they permit the classification of flow conditions based on their dominant dynamic features. The flow regimes are treated separately to emphasize their physical characteristics, assumptions of the governing equations, and mathematical modeling formulations. After Karney (1990), transient flow can be classified in the following regimes:

- Unsteady-compressible flow,
- Unsteady-incompressible flow,
- Quasi-steady flow, and
- Steady-state flow.

The key flow characteristics, corresponding modeling approaches, and underlying physical assumptions are discussed below, as well as how these relate to the governing equations.

2.2.1 Unsteady-Compressible Flow

Under unsteady-compressible pipe network flow, the hydraulics are highly unsteady. Such conditions are characterized by rapid exchanges of kinetic and potential energy between the fluid and conduit: this appears in a system as high-frequency pressure and flow oscillations. Compressibility effects, inertial effects, and possibly even unsteady friction are usually apparent. An example of unsteady-compressible flow conditions are those in a pumping pipeline following
a power failure. Rapid flow deceleration occurs due to the pump’s sudden loss of power, and a negative pressure wave initially propagates away from the source. The wave is then modified by friction and reflects off of other boundaries. As energy dissipates through viscous fluid resistance, the flow conditions will eventually reach a new steady state. Such conditions are invoked by large changes in a network’s boundary conditions over a relatively short period of time.

Water hammer models are used to simulate unsteady-compressible flow. These models consider compressibility effects, inertial effects, fluid resistance, and even unsteady fluid resistance. The governing PDEs (Equations [2.4] and [2.5]) are solved with the reasonable assumption that the advective terms \((Q/A) \cdot \partial Q/\partial x\) and \(Q \cdot \partial H/\partial x\), as well as \(Q \cdot \sin(\alpha)\), are negligible. From this, Equations [2.4] and [2.5] reduce to (Chaudhry, 2014)

\[
\frac{1}{gA} \frac{\partial Q}{\partial t} + \frac{\partial H}{\partial x} + J_{QF} + J_{UF} = 0
\]

and

\[
\frac{a^2}{gA} \frac{\partial Q}{\partial x} + \frac{\partial H}{\partial t} = 0
\]

Water hammer models are capable of simulating all of the one-dimensional unsteady flow regimes. However, their greater physical accuracy and complexity are not always warranted, for water hammer models are accompanied by greater computational demand due to their required use of a smaller time step and spatial discretization.

In practice, water hammer models are used to assess the severity of critical transient events. Their results are used to design protective measures that mitigate undesirable consequences. Water distribution systems often span large distances and have complex topologies; consequently, they are host to an array of flow conditions due to frequent controlled and uncontrolled operating changes. Hydro power facilities are also highly dynamic. They are subject to risk due to having long penstocks with high inertia. Because both of these types of systems have high potential for generating unsteady-compressible flow conditions, water hammer models are frequently used to ensure their structural integrity.

### 2.2.2 Unsteady-Incompressible Flow

At modest degrees of unsteadiness, compressibility effects are less apparent. Under unsteady-incompressible flow, inertial effects are the dominant dynamic flow characteristic, and flow disturbances propagate much faster than the duration of the transient-initiating event. Such
conditions are generated by events of small magnitude, which are most often controlled operations: for example, the closure of a valve over a relatively long period of time, or a controlled pump start. It is difficult to define what constitutes a sufficiently long period of time, for this is subjective and unique to each network.

Rigid water column (RWC) models are well suited to simulating unsteady-incompressible flow. In addition to the assumptions made by water hammer models, these models also assume that changes at a network’s boundaries occur slowly enough such that the fluid behaves as an incompressible body. By rewriting Equation [2.7] as

\[ \frac{gA}{a^2} \frac{\partial H}{\partial t} + \frac{\partial Q}{\partial x} = 0 \]  

[2.8]

it can be seen that as the time rate of change in a pipe’s boundary conditions, \( \partial H / \partial t \), becomes small, the spatial variation in flow along a pipe’s length, \( \partial Q / \partial x \), also tends to zero, which is characteristic of incompressible flow. This leaves only the momentum equation, a single ordinary differential equation (ODE), to be solved.

Under unsteady-incompressible flow, integrating the momentum equation over the length of a pipe leads to

\[ \frac{L}{gA} \left( \frac{dQ}{dt} \right) + F(Q) - H_1 + H_2 = 0 \]  

[2.9]

where \( H_1 \) and \( H_2 \) = the heads (m) at the upstream and downstream ends of the pipe, respectively, and \( F(Q) \) = the head loss (m), which encompasses friction losses and head gains (e.g., from pumps). Equation [2.9] is the unsteady-incompressible momentum equation. Its left-most term, sometimes referred to as the acceleration head, represents the importance of inertial effects: if it is small, inertial effects are negligible. However, if the acceleration head is large, compressibility and possibly even unsteady friction effects may also be present. As is the case with fast and slow transients, “small” and “large” are relative terms that are difficult to define.

Intermediate RWC models have seldom been used. These models have been shown to adequately simulate controlled operations, but their use is inhibited by the perception that they have little practical utility due to neglecting compressibility effects. Nevertheless, RWC models do not require spatial discretization of pipes (and are therefore often referred to as lumped models), and they use a modest time step (on the order of \( 10^{-1} \) to \( 10^{1} \) s); accordingly, they are more computationally efficient than water hammer models. Despite this, the practical use of RWC models has been limited by a poorly understood range of validity.
2.2.3 Quasi-Steady and Steady-State Flow

For progressively smaller magnitude and slower boundary condition changes, unsteady flows enter the quasi-steady flow regime and ultimately the steady-state flow regime. In both cases, head losses are the dominant flow characteristic. The distinguishing feature between quasi-steady and steady-state conditions is that a network’s hydraulics change slowly over time for the former (i.e., $\partial Q/\partial t \approx 0$ and $\partial H/\partial t \approx 0$), while they are time-invariant for the latter (i.e., $\partial Q/\partial t = 0$ and $\partial H/\partial t = 0$). Accordingly, the approximation $\partial Q/\partial t = 0$ is made for the momentum equation, even though $\partial Q/\partial t$ is not strictly zero under quasi-steady conditions. It is worth noting that the above representations are only mathematically artefacts; in reality, steady flow conditions arise when the time-averaged variation in flow is near zero.

Physically, quasi-steady flow conditions arise when small-magnitude boundary changes occur over long periods of time. Comparatively, steady-state conditions arise when a network’s boundary conditions remain fixed, and the hydraulics become coupled into a state of network-wide equilibrium. Quasi-steady and steady-state flow models, name respective of their simulated conditions, consider slightly different flow conditions, yet both assume that compressibility and inertial effects are negligible. In addition to the assumptions made by water hammer and RWC models, these models assume that $\partial Q/\partial t$ is approximately zero: from this, the unsteady-incompressible momentum equation (Equation [2.9]) reduces to

$$F(Q) - H_1 + H_2 = 0$$  \[2.10\]

Regarding implementation, quasi-steady models simulate non-inertial unsteady flow using a series of linked steady-state solutions. In practice, such models are frequently used to perform extended period simulations (EPSs), often to assess a network’s performance over the course of hours or even days. Within such simulations, inertial and compressibility effects are intentionally neglected, even if they are in fact present from time to time.

2.2.4 Flow Classification

From the preceding sections, each transient flow regime features a distinct combination of dynamic effects. Figure 2.1 illustrates the relative importance of inertial and compressibility effects as a function of unsteadiness. The curves in Figure 2.1 are only qualitative, but they illustrate how the relative importance of each dynamic term varies with the degree of unsteadiness. Note that steady-state flow conditions are represented at an abscissa value of zero.
Chapter 2: Transient Flow Modeling

Classifying unsteady flows is not a completely straightforward task. This is because the boundaries between the unsteady flow regimes are not always clear, and they are often unique to each network. In studying the transient flow regimes, Karney (1990) derived an integrated energy equation for transient flow that represents the governing momentum and continuity equations. The energy equation for unsteady-compressible flow for an individual pipe is (Karney, 1990)

\[
\frac{dU}{dt} + \frac{dT}{dt} + D' + W' = 0
\]  

where \( U \) = the internal energy (J), \( T \) = the total kinetic energy (J), \( D' \) = the rate of viscous dissipation (J/s), and \( W' \) = the rate at which work is done to force the fluid through the pipe (J/s). The terms in Equation [2.11] are respectively given by

\[
\frac{dU_j}{dt} = \frac{\rho A_j}{2} \left( \frac{\partial}{\partial t} \right)^2 \int_{x=0}^{L_j} (H_j)^2 \, dx
\]  

\[
\frac{dT_j}{dt} = \left( \frac{\rho}{A_j} \right) \int_{x=0}^{L_j} \left( \frac{\partial L_j}{\partial t} \right)^2 \, dx
\]  

\[
D_j' = \frac{\rho g}{L} \int_{x=0}^{L_j} F(Q_j) Q_j \, dx
\]  

\[
W_j' = \rho g (Q_2 H_2 - Q_1 H_1)
\]

where the subscripts 1 and 2 respectively refer to the start and end nodes of the pipe. Note that the equations above differ from those presented by Karney (1990) in that they are expressed in terms of flow rather than velocity and the head loss term has a more general representation.
To classify unsteady flows, Karney (1990) suggested comparing the relative importance of the equation’s components. For example, if the internal energy term $dU/dt$ is relatively large compared to the other terms in Equation [2.11], then compressibility effects are present. From this, Karney proposed the following compressibility index for assessing the importance of compressibility effects:

$$\phi = \frac{|\Delta U|_{\text{max}}}{|\Delta T|_{\text{max}}}$$

where $|\Delta U|_{\text{max}}$ = the greatest change in internal energy (J), and $|\Delta T|_{\text{max}}$ = the greatest change in total kinetic energy. The compressibility index provides a means of characterizing the largest magnitude of kinetic-internal energy transformations, but it requires that a simulation be performed in advance using a water hammer model.

### 2.3 Pipe Network Modeling Developments

Developments in multi-flow regime modeling are reviewed here. The literature review is divided into three parts: steady-state and quasi-steady flow modeling; unsteady-incompressible flow; and characterizing transient flows and developing advanced adaptive models. First discussed are key developments regarding modeling steady-state and quasi-steady flows, and the second section provides a detailed review of RWC models in the literature. The latter directly relates to the first of this thesis’ objectives. Lastly, the third section addresses this thesis’ central objective: it focuses on studies pertaining to delineating the transient flow regimes.

#### 2.3.1 Water Hammer Models

Many numerical techniques have been developed to solve Equations [2.6] and [2.7]. The most common is the method of characteristics (MOC), which has proven reliable, easy to implement, and it accurately captures shock fronts. Alternatives include finite difference (e.g., Chaudhry and Hussaini, 1985), finite volume (Guinot, 2000; Zhao and Ghidaoui, 2004; León et al., 2008), finite element (e.g., Bisgard et al., 1987), and other characteristic-type solutions (Wood et al., 1966; Jung et al., 2009). An overview of these can be found in Ghidaoui et al. (2005) and Wood et al. (2005). Each approach solves the same governing equations, but through different means.

To track wave fronts, spatial discretization is required. This involves subdividing individual pipes into $N$ interior reaches, each of length $\Delta x = L/N$ (m; Figure 2.2). To obtain a numerically stable solution, pipe discretization must satisfy the Courant-Friedrichs-Lewy (CFL) criterion $C =$
$a \cdot \Delta t/\Delta x \leq 1$, where $\Delta t$ is the time step (s). For pipe networks, this presents a key challenge: a common $\Delta t$ that satisfies the CFL criterion for all pipes is required.

![Figure 2.2: Example pipe network showing base network and discretized reaches](image)

Networks may comprise hundreds to thousands of pipes. Each can have different properties, so it is difficult to determine $\Delta t$ that simultaneously satisfies the CFL criterion for each pipe. Approaches to this include pipe length adjustment, wave speed adjustment, and interpolation techniques (e.g., Lai, 1982; Goldberg and Wylie, 1983; Sibethers et al., 1991; Karney and Ghidaoui, 1997) – errors associated with the latter have been studied extensively (e.g., Ghidaoui and Karney, 1994; Shimada et al., 2006; Shimada et al., 2008). In addition to satisfying the CFL criterion, $\Delta t$ on the order of $10^{-1}$ to $10^{-2}$ s are often necessary to resolve highly unsteady flows. The challenge is that this yields a greater number of time steps and a greater number of interior pipe nodes, making the choice of $\Delta t$ key to computational efficiency.

Water hammer models are more complex and computationally demanding than their one-dimensional counterparts. Despite this, RWC and quasi-steady models are incapable of simulating compressibility effects and unsteady friction, so water hammer models are unique in this manner. The greater accuracy of water hammer models, however, is also not always required.

### 2.3.2 Unsteady-Incompressible Flow Models

Unlike the other transient flow regimes, unsteady-incompressible flow has been relatively neglected in the literature. There have been modeling advances, yet none are recent. Nonetheless, RWC models, which simulate unsteady-incompressible flow, are an efficient alternative to water hammer models that thus merit attention.

Both loop- and node-based RWC formulations have been developed for simulating unsteady-incompressible flow in arbitrary pipe networks. Holloway (1985) and Onizuka (1986) were the
first to present loop-based mathematical formulations. The former solved the governing unsteady-incompressible momentum equation for pipe network loops, which was accomplished using a semi-implicit linearized representation of the non-linear head loss term. In comparing simulation results with those of a water hammer model, there was good agreement when simulating moderately unsteady conditions, such as those arising from controlled operations; accordingly, RWC models were shown be a practical alternative for such conditions. Like Holloway, Onizuka (1986) also developed a loop-based RWC formulation. However, rather than linearizing the non-linear \( F(Q) \) term, Onizuka solved a system of first order ODEs using an explicit numerical method. Additionally, a Lyapunov function was derived to describe the degree of unsteadiness, and the author discussed a dynamic relaxation technique for steady-state analyses.

Islam and Chaudhry (1998) later combined Holloway’s RWC formulation with a water quality model. The authors investigated whether dispersion and decay differ significantly when inertial effects are considered, and in contrasting their results against those of a quasi-steady model, little difference was observed. Though, the authors tests are limited to a small looped pipe network, which is not representative of any realistic system. Further, unsteady flow models more so pertain to hydraulic aspects, namely pressures.

As an alternative to the loop-based formulations, Shimada (1989) introduced the incidence method by using incidence matrices to represent arbitrary network topologies. It is superior to the loop-based methods for two reasons: pseudo loops and loop identification are not required, and changes in network topology (i.e., due to opening and closing links) is easily accommodated. While advantageous, numerical instabilities are problematic due to the use of explicit numerical integration to solve a system of stiff differential equations. \textit{Stiff differential equations} exhibit relatively largely varying rates of change, so they are difficult to integrate numerically using explicit methods. Here, stiffness challenges are problematic due to a wide range of link resistances and zero flows, which arise from the prevalence of quasi-steady conditions and closing links (Axworthy, 1997). To resolve the matter, Shimada suggested using higher order explicit methods, though the numerical instabilities persisted.

Shimada (1992) later improved the incidence method with the use of state-space notation. The approach was re-formulated using state variables (i.e., the minimum number of variables required to describe a system’s behavior) to improve computational efficiency; however, the numerical instabilities still persisted. Axworthy and Karney (2000) further improved the incidence method by developing an algorithm for simulating full valve closure. Unlike previous approaches,
Chapter 2: Transient Flow Modeling

which involve either assigning a non-zero final discharge or distributing residual flows across a network when the valve becomes fully closed, their algorithm allows the incidence method to compute a final valve discharge of zero upon full closure. Axworthy and Karney also compared their RWC simulation results with those of a water hammer model and found that both models’ results agreed up until the time when the valve became fully closed. After this point, RWC formulations naturally fail to capture residual waves resulting from compressibility effects.

More recently, Ahmed (1997) developed an RWC formulation with a structure similar to that of the global gradient algorithm (GGA; Todini and Pilati, 1988). The GGA is a proven efficient and robust steady-state solver. Ahmed’s formulation has similar traits, yet, like Holloway (1985), the author linearized the headloss term, thereby sacrificing its integrity. Additionally, only a fixed discretization of the momentum equation was considered, which may otherwise aid solution stability, as is the case for analyses of networks with tanks (Todini, 2011).

While each of the aforementioned studies constitutes an important development, the numerical instabilities of RWC models have persisted. Additionally, there are other key challenges that remain unsolved, including the following: RWC models have yet to be applied to large, complex pipe networks; there are no clear guidelines for selecting an appropriately sized time step size to ensure stability and provides adequate numerical accuracy; and the range of validity of quasi-steady and RWC models are not well articulated. These motivate the objectives of this thesis that concern developing improved RWC formulations that are robust, stable, and accurate. Further discussions of literature relating to the latter gap (i.e., the range of validity of quasi-steady and RWC models) are considered later on in this chapter.

2.3.3 Quasi-Steady and Steady-State Flow Models

Quasi-steady and steady-state models are widely used. Their applications include short- and long-term capacity analyses (i.e., EPSs), evaluating water quality, and network optimization. The terms “quasi-steady model” and “EPS” are often used interchangeably, but the former is a type of model and the latter an application thereof. Due to their wide use, steady-state models have received considerable attention in the literature. It is impractical to review all key developments, so only key studies are discussed. The reader is referred to Ormsbee (2006) and Todini and Rossman (2013) who summarize the key historical steady flow modeling developments.

One of the most significant developments regarding modeling steady-state hydraulics is the GGA (Todini and Pilati, 1988). Also referred to as the Newton-Raphson global algorithm, the
Chapter 2: Transient Flow Modeling

GGA uses the Newton-Raphson method to solve the steady-state momentum and nodal continuity equations for arbitrary pipe networks by minimizing residual differences. The GGA has proven to be a robust formulation that incorporates various boundary conditions (including variable-speed pumps, control valves, and pressure-dependent outflows), and it does not require initial solution estimates that satisfy nodal continuity. Further, it has favourable convergence properties (Todini and Rossman, 2013). Owing to the aforementioned, many modern software packages use the GGA (e.g., EPANET 2; Rossman, 2000).

Numerous developments followed the introduction of the GGA. For example, various studies have considered pressure-dependent outflows (e.g., Ang and Jowitt, 2006; Giustolisi et al., 2008; Wu et al., 2009; Giustolisi and Laucelli, 2011), handling zero flows (Elhay and Simpson, 2011), and considering higher order solution terms (Simpson and Elhay, 2011). Some studies have even introduced new formulations and compared them against the GGA (e.g., Todini and Rossman, 2013; Creaco and Franchini, 2014; Elhay et al., 2014), though the latter remains the preferred alternative due to its favourable properties.

As an extension to steady-state models, quasi-steady models (also referred to as quasi-static or non-inertial unsteady models) link a series of steady-state solutions. Rao and Bree (1977) and Rao et al. (1977) are attributed with being some of the first to present a formal quasi-steady modeling approach: the authors coupled the explicit modified Euler method with a steady-state formulation to update tank water levels at the end of each time step. Iterations are performed until the “trial” and “final” tank water level estimates converged within a specified tolerance. To minimize numerical error, nodal mass balance residuals are allocated to a network’s reservoirs and tanks. Bhave (1988) improved this approach by directly incorporating the nodal continuity for tanks into the solution, thereby resulting in an implicit solution for tank continuity.

Despite the above, commercial quasi-steady models still use the simpler Euler method for integrating the nodal mass conservation equation for tanks. Numerical oscillations were later found to be problematic for networks having tanks with highly variable levels (Todini, 2011), the result of using an explicit solution. To overcome this, Todini (2011) extended the GGA to simulate quasi-steady flow with variable implicit integration, which mitigated the numerical instabilities. Avesani et al. (2012) implemented this within EPANET (Rossman, 2000), and Giustolisi et al. (2012) later extended Todini’s (2011) work to consider pressure-dependent outflows, thus forming the generalized GGA (G-GGA). These represent the key quasi-steady and steady flow modeling developments to date.
2.4 Flow Regime Delineation and Adaptive Modeling

Water hammer, RWC, and quasi-steady models are discussed separately in the previous section. Unsteady flows are not, however, strictly confined to a single flow regime; rather, long- and even short-term hydraulics never belong to a single flow regime. Unsteady flows thus transition between the regimes. In practice, only one type of model is used for a given application. Consequently, only its corresponding flow regime is considered as an upper bound for physical accuracy (for example, an RWC model’s range encompasses that of quasi-steady models). One concept that seeks to resolve this is adaptive hybrid modeling. Within the context of unsteady flow modeling, hybrid models combine different types of models, and the adaptive component switches between the solvers. The high-level objective of adaptive hybrid modeling is to only use those less efficient, more complex solvers when their accuracy is needed. Otherwise, a more efficient solver is used. The main challenge with adaptive hybrid modeling is understanding when to use which type of solver, a task that is analogous to the model selection task, itself unresolved.

Many studies have investigated characterizing the transient flow regimes. Early efforts were motivated by the need to justify approximate analyses due to the lack of computers and later computing power. Two notable studies are Joukowsky (1898) and Allievi (1915), each of whom introduced namesake indicators (the fundamental Joukowsky equation and the Alleivi parameters). Others sought to resolve the transient flow characterization matter (Parmakian, 1955; Valentine, 1965), yet conclusive findings have remained elusive. Of course, characterizing transient flows is no simple task. In addition to different boundary condition changes, every pipe network is unique. To provide context to the current work, more recent studies that have investigated the transient flow regimes are discussed below.

In deriving the integrated energy equation for unsteady-compressible flow, Karney (1990) suggested comparing the equation’s terms to classify transient flow conditions. To this extent, a compressibility index was proposed for assessing the importance of compressibility effects. The index represents the ratio of the maximum changes in internal and kinetic energies, and though it was shown to be insightful, the indicator suffers two limitations. It only describes one of the two flow regime boundaries, and applications to large, complex pipe networks were not considered.

Wood et al. (1990) studied transient flow with the objective of discerning when to apply water hammer or RWC models. The authors developed the following paraphrased qualitative criterion:

For fast transient events, water hammer models are required to accurately simulate pressures and flows. An event is “fast” if it occurs on a time scale that is short compared
to the characteristic time of a system. For slow varying flows, a quasi-steady model is sufficient. When the event is moderately fast yet slow relative to a system’s characteristic time, an RWC model may be used.

Indeed, this is an accurate but not too helpful observation. The terms “fast” and “slow” are relative and ambiguous, and they do not explicitly prescribe the conditions under which a water hammer, RWC, or quasi-steady model should be employed. Rather, the above is merely a restatement of the problem at hand. Cabrera et al. (1995) also investigated the conditions under which the different models should be employed, but they arrived at a similar qualitative conclusion to Wood et al. (1990).

Rogalla and Wolters (1994a; 1994b) took a different approach. A hybrid formulation was developed by combining an MOC-based water hammer model with an implicit finite difference scheme for pipes represented using an RWC model. The latter alleviates the CFL criterion by allowing some internal pipe reaches to be modeled with incompressible flow. As a result, some pipe reaches are assigned unrealistic wave speeds, which may skew or alter simulation results. The hybrid model was shown to have greater computational efficiency than an MOC-based water hammer model, yet the authors’ work uses unrealistic wave speeds which may skew simulation results. Additionally, Rogalla and Wolters did not investigate when an RWC model becomes more suitable than a water hammer model.

In a similar yet more comprehensive manner than Wood et al. (1990), Abreu et al. (1999) studied the conditions under which each transient model should be applied. The authors reformulated the governing equations in terms of dimensionless variables, and this lead to the development of two transient flow indicators. These were demonstrated for relatively simple pipe networks. While detailed, the findings closely resemble those of previous studies. Furthermore, the authors’ approach requires a priori knowledge of a system’s unsteady hydraulics, and only relatively simple pipe networks were considered.

Each of the above studies aimed to characterize unsteady flows post-simulation. Instead, Axworthy (1997) presented a comprehensive transient pipe network model capable of simulating the full range of unsteady flows, one that reactively adjusts itself according to the simulated conditions. Two key contributions of the author’s work are formulating boundary conditions that can interface with different types of models and developing a comprehensive unsteady flow model. For transitions between the RWC and quasi-steady solvers, two criteria were used: one for switching from the quasi-steady to RWC solver, and another for switching back. Because
Axworthy used Shimada’s (1989) incidence method with explicit numerical integration, both criteria were based on the former’s stability. Additionally, the use of explicit numerical was required to couple the water hammer and RWC solvers. Altogether, Axworthy’s (1997) model performed well for a limited number of test cases, but there are numerical challenges.

Like Axworthy (1997), Filion and Karney (2002) also developed a hybrid transient model. The authors combined water hammer and quasi-steady solvers, and the model was developed for EPSs. Two levels of simulations were proposed: water hammer microsimulations were implemented with a micro time step whenever a boundary change occurred, and extended period quasi-steady simulations were performed between pre-set intervals. This allowed the model to accurately simulate pump and valve operations within EPSs. Although shown to perform well, the authors’ work suffers two key limitations: a fixed front-end duration must be prescribed, and the model is computationally intensive. The former is particularly problematic, for a priori knowledge of the network’s unsteady hydraulics is necessary. If too large of a duration is used, the scheme has a large run time; conversely, if too short of a duration is used, then key transient hydraulics are overlooked. Despite these, Filion and Karney demonstrated their model’s ability to provide a better resolution than conventional EPSs as well as improved reservoir routing. Furthermore, hybrid modeling was shown to aid the identification of critical transient states: typical transient models are incapable of this, for they require prescribed initial conditions.

All of the above-noted literature share common limitations. Current transient flow indicators have only been tested with simple pipe networks, and studies predominantly consider valve operations. Other challenges include balancing the competing demands of physical accuracy and computational efficiency, both of which are essential to practical analyses. These are the motivation for this thesis’ contributions.

2.5 Example – Valve Closure in a Simple System

To illustrate how each model simulates unsteady hydraulics for different degrees of unsteadiness, consider the simple reservoir-valve-reservoir network in Figure 1.2. Valve V1 is initially fully open, with an initial steady discharge of 0.21 m³/s and velocity of 3.0 m/s. The partial closure and subsequent opening of valve V1 induce transient conditions, and the degree of unsteadiness directly depends on how fast the operation is performed. Figure 2.3 depicts the simulated head for two events (valve closure and valve opening) and different operation durations.

Figure 2.3 shows a number of similarities and differences amongst the models. All three
models simulate the same initial and final heads when the flow conditions are steady, regardless of the valve closure duration; however, the transient hydraulics between the initial and final states differ slightly to significantly depending on the degree of unsteadiness. This is due to the underlying physical assumptions of each model.

Figure 2.3: Reservoir-valve-reservoir system

For a valve operation duration of 500 s, the results within Figure 2.3a and b are practically indistinguishable. This is because the operation is relatively slow, and though the results are similar, the water hammer and RWC models require significantly more computational effort. For a shorter operation duration of 5 s, each model predicts different behaviour. Figure 2.3c and d show that the water hammer model adequately captures the resulting dynamic effects, yet the quasi-steady model fails to simulate even the inertial effects that are captured by the RWC model. Amidst these two extremes, a valve operation period of 50 s produces modest unsteady flow conditions whereby inertial effects are the dominant flow mechanism.

Altogether, each model is suited to a different range of flow conditions. A water hammer model, while capable of accurately simulating each event, is not necessary for slower operations, whereas the simpler quasi-steady model is restricted to simulating such conditions. It is thus imperative to consider a model’s physical assumptions and complexity, particularly how they limit its practical use.
Chapter 2: Transient Flow Modeling

Figure 2.4: Junction J1 head traces for various valve operations

(a) 75% uniform valve closure over 500 s
(b) 75% uniform valve opening over 500 s
(c) 75% uniform valve closure over 50 s
(d) 75% uniform valve opening over 50 s
(e) 75% uniform valve closure over 5 s
(f) 75% uniform valve opening over 5 s
2.6 Summary

Unsteady flow modeling was reviewed in this chapter. Beginning with the governing equations, the underlying flow physics were discussed within the context of the transient flow regimes and parallel modeling approaches, two central themes to the current work. A review of recent modeling developments highlighted three gaps in the literature. First, current RWC formulations suffer numerical limitations, yet there have been no recent efforts to address this. Numerous studies have also sought to characterize transient flows; however, conclusive findings have yet to be obtained. Finally, past efforts to balance the accuracy-efficiency tension are limited by numerical matters, inefficient solvers, and sub-optimal adaptive schemes.

Further to the above, the tension between physical accuracy and computational efficiency remains unresolved. The importance of this was illustrated by simulating the unsteady hydraulics of a sample pipe network: results showed how each model is suited to a different range of unsteady flow conditions. Overcoming these limitations is a necessary step towards improving transient modeling practices. Ultimately, it would prove beneficial to develop a comprehensive adaptive hybrid model similar to that of Axworthy (1997) or Filion and Karney (2002), yet with refined unsteady flow indicators, a more robust solver, and testing with large networks. These topics are all addressed in the current work.
Chapter 3: Pipe Network Representation

At a basic level, pipe networks can be represented by interconnected nodes and links. Networks vary largely in both configuration and size, and some models comprise tens of thousands of elements. Accordingly, there is a need to represent the topology (i.e., node-link interconnections) of arbitrary systems in a simple yet comprehensive manner. Fortunately, graph theory provides exactly this. When combined with concepts from linear algebra and calculus, graph theory facilitates the mathematical computations necessary for hydraulic simulation.

Here, key concepts from graph theory are discussed as they relate to pipe network analyses. Central concepts are reviewed first from Strang (1987), who provides a descriptive overview of graph theory and its role in applied mathematics. Relationships between node-link connections and graph loops are then discussed in §3.2, and, from these, key relations are illustrated for an example system. Using graph theory notation, §3.4 formulates network conservation equations for mass and momentum: these describe incompressible network hydraulics. Along with modifications introduced later in this thesis, they also form a key part of unsteady-compressible flow models. This chapter concludes with a review of recent literature in order to relate the fundamentals to the state-of-the-art.

3.1 Fundamentals of Graph Theory

Many systems comprise sets of nodes that are interconnected by various pathways. For example, transportation networks can be thought of as a set of destinations linked by roadways through which people travel, while electrical distribution grids consists of transmission lines that convey electricity to various consumers. Each can be represented by a graph. A graph is a network of nodes interconnected by links, and graph theory is the study of graphs and problems involving graphs. Here key concepts and definitions are reviewed.

3.1.1 Directed Graphs and Incidence Matrices

A graph $G$ is defined by its nodes and the links that interconnect them. Nodes are a point at which one or more links meet, and they are denoted by $N(G) = \{N_1, ..., N_{n_N}\}$ for a graph $G$, where $n_N$ = the number of nodes. Similarly, $L(G) = \{L_1, ..., L_{n_L}\}$ represents the collection of a graph’s $n_L$ links. A graph may also contain reference nodes that serve as a datum or source.

Consider the graphs below from Strang (1986; Figure 3.1). Both graphs $G_1$ and $G_2$ have the
same nodes and links, so \( N(G_1) = N(G_2) \) and \( L(G_1) = L(G_2) \), where node \( N_1 \) is a reference node. Graph \( G_1 \) (Figure 3.1a) is an **undirected graph**, for its links have no specified direction. Conversely, each of graph \( G_2 \)'s (Figure 3.1b) links has a direction, making it a **directed graph**.

![Graphs](image)

### Figure 3.1: Undirected and directed graphs (Strang, 1987)

Node-link interconnections between a directed graph’s nodes and links can be described by a **node-link incidence matrix** \( A \in \mathbb{R}^{n_N \times n_L} \). The elements of \( A \) are given by

\[
A_{ij} = \begin{cases} 
+S_j & \text{if link } j \text{ starts at node } i \\
0 & \text{if link } j \text{ is not connected to node } i \\
-S_j & \text{if link } j \text{ ends at node } i 
\end{cases} \tag{3.1}
\]

where \( S_j = \) the status of link \( j \) given by

\[
S_j = \begin{cases} 
1 & \text{if link } j \text{ is open} \\
0 & \text{if link } j \text{ is closed} 
\end{cases} \tag{3.2}
\]

The node-link incidence matrix for the directed graph \( G_2 \) (Figure 3.1b) is

\[
A = \begin{bmatrix} 
+1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & +1 & +1 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & +1 & +1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 0 & +1 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & -1
\end{bmatrix}
\]

When each node is connected to each other node through at least one path, a graph is said to be **connected**. A **path** is a unique series of alternating nodes and links that starts and ends at a node. Paths may be open or closed: an open path has no loops and starts and ends at different
nodes, while a closed path or loop starts and ends at the same node. As an example, nodes $N_1$ and $N_4$ in graph $G_2$ can be connected by an open path comprising the nodes $\{N_1, N_3, N_2, N_4\}$ and links $\{L_2, L_3, L_4\}$. The nodes $\{N_1, N_2, N_5, N_3, N_1\}$ and links $\{L_1, L_5, L_6, L_2\}$ form a closed path. A set of graph $G_2$’s loops are shown below (Figure 3.2a).

![Graph G2 and G3](image)

(a) Graph $G_2$ with an incomplete set of loops  
(b) Graph $G_3$ with a set of complete loops

Figure 3.2: Incomplete and complete sets of loops

A set of the $n_l$ loops describing a graph $G$ are denoted by $l(G) = \{l_1, \ldots, l_{n_l}\}$. Similar to a node-link incidence matrix, a graph’s topology can be described using the loop-link incidence matrix $B \in \mathbb{R}^{n_l \times n_L}$. It relates a graph’s loops and links, and the elements of $B$ are given as

\[
B_{kj} = \begin{cases} 
-S_j & \text{if edge } j \text{ has the opposite direction to loop } k \\
0 & \text{if edge } j \text{ is not in loop } k \\
+S_j & \text{if edge } j \text{ has the same direction as loop } k 
\end{cases} \quad [3.3]
\]

Loop directions can be arbitrary. The loop-edge incidence matrix for graph $G_2$ (Figure 3.2a) is

\[
B = \begin{bmatrix}
-1 & +1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & +1 & 0 & -1 & +1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 1 & -1
\end{bmatrix}
\]

### 3.1.2 State Variables

Some interesting relationships can be observed by comparing $A$ and $B$. One concerns state variables – these represent comprise a set of the minimum number of variables required to fully describe flows through all of the graph’s links and the potential (e.g., energy) at its nodes. If too few variables are used, then the graph cannot be described completely. More variables than the state variables may be used, though this leads to redundant variables.
If \( A \) is contrasted against \( B \) for \( G_2 \), both describe the flows through graph \( G_2 \)’s links (Figure 3.2a). However, each column of \( A \) sums to zero, implying that some rows are redundant, so the columns of \( A \) are \textit{linearly dependent} (i.e., some columns are linear combinations of other columns). While not problematic, \( A \) provides more information than is needed. Accordingly, it is not the most efficient means of describing \( G_2 \). Certain rows can be eliminated to reduce \( A \), and those rows can be reconstructed from the remaining rows. The loop-link incidence matrix \( B \) has its own shortcoming. It describes graph \( G_2 \) with fewer rows than \( A \); however, \( B \) is unable to describe link \( L_4 \) other than that it is not within any loops.

In order to fully describe graph \( G_2 \) via loops, the pseudo link \( L_9 \) is introduced (Figure 3.2b). This link is not real, but rather a fictitious connection that permits \( B \) to fully describe the modified graph \( G_3 \). The incidence matrices for graph \( G_3 \) are

\[
A = \begin{bmatrix}
+1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & +1 & +1 & +1 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & +1 & +1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & +1 \\
0 & 0 & 0 & 0 & -1 & -1 & 0 & +1 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
-1 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & +1 & 0 & -1 & +1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & -1 & +1 & -1 & 0 & 0 \\
\end{bmatrix}
\]

All flows in graph \( G_2 \) can be described using the loop-link incidence matrix \( B \) of graph \( G_3 \). Consequently, the rows of \( B \) are \textit{linearly independent}, and the loops \( \{l_1, l_2, l_3, l_4\} \) constitute a set of state variables for graph \( G_3 \). Coincidently, the number of state variables is equivalent to the number of loops. There are numerous sets of independent loops: one loop may be modified by joining it with an adjacent loop (the numerical equivalent comprises adding the corresponding rows). Unless a graph has a single loop, there are multiple unique complete loop sets.

### 3.1.3 Graph Partitioning

Graphs can be partitioned in multiple ways according to their topology. Often a graph is partitioned into two components, one that connects to all of its nodes and another that complements this. Another approach is to divide a graph between its interior loops and exterior branches. These graph partitions relate to how we analyze and partition networks.

Recall from §3.1.1 that a path traverses a set of nodes and links. A set of connected open
paths with no loops constitutes a tree, and a tree that traverses all nodes within a graph is a spanning tree or dendrite. A spanning tree for a graph with \( n_R \) reference nodes has \( n_B = n_N - n_R \) links: those \( n_B \) links are called tree branches, while those not in the spanning tree are called chords. The chords form a co-tree that complements the spanning tree. Depending on its topology, a graph may also have multiple spanning trees (Figure 3.3).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{spanning_trees.png}
\caption{Two spanning trees for graph \( G_2 \) (Strang, 1987)}
\end{figure}

Within Figure 3.3a, links \{\( L_1, L_2, L_4, L_5, L_7 \)\} form the tree branches of one spanning tree. Complementary to that, the links \{\( L_3, L_6, L_8 \)\} form the chords of the co-tree. The columns of \( A \) for the graph in Figure 3.3a can be reordered as \{\( L_1, L_2, L_4, L_5, L_7, L_3, L_6, L_8 \)\} to partition the incidence matrix between the spanning tree branches and chords as

\[
A = [A_t | A_c] = \begin{bmatrix}
+1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & +1 & 0 & +1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & +1 & -1 & +1 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & +1 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0
\end{bmatrix}
\]

where \( A_t \in \mathbb{R}^{n_N \times n_B} \) and \( A_c \in \mathbb{R}^{n_N \times n_C} \) are the tree and co-tree incidence matrices. If at least one spanning tree exists, then a graph is connected, for the spanning tree traverses all nodes.

Some key relations can be drawn from the above definitions. Consider a connected graph with \( n_N \) nodes, \( n_L \) loops, \( n_L \) links, and \( n_R \) reference nodes. Any spanning tree of this graph must have \( n_B = n_N - n_R \) branches. The total number of links is \( n_L = n_B + n_C \), so the number of chords is \( n_C = n_L - n_N + n_R \). Since each chord added to a spanning tree forms a loop (e.g., in
Figure 3.3a, adding the chord $L_3$ to the spanning tree forms a loop with links \{L_1, L_2, L_3\}, it follows that a graph must have $n_l = n_L - n_N + n_R$ independent loops. This can be observed for the spanning trees in Figure 3.3: the number of loops is $n_l = n_L - n_N + n_R = 8 - 6 + 1 = 3$ loops, which agrees with the three loops illustrated in Figure 3.2a.

A graph can also be partitioned between its external forest and core. A graph’s core comprises its interior looped blocks, and its forest includes the remaining external tree branches. An example is illustrated below (Figure 3.4). The forest-core concept distinguishes between graph segments that concern nodal balances (i.e., the forest) and those which for loop balances (i.e., the core).

In addition to paths, trees, and loops, a graph itself can contain other graphs. A graph $G_S$ is a subgraph of graph $G_D$ if $N(G_S) \subseteq N(G_D)$ and $L(G_S) \subseteq L(G_D)$, $N(G_S)$ contains at least one reference node from $N(G_D)$, and $L(G_S)$ and $L(G_D)$ have the same direction. Spanning trees and cores themselves are in fact subgraphs. An example of a subgraph is shown below (Figure 3.5).
3.1.4 A Brief Summary

A number of fundamental concepts from graph theory were illustrated in this section. The key points are summarized below:

- **Directed and undirected graphs**: the links within a directed graph have specified start and end nodes. Comparatively, those in an undirected graph are direction-less. Graphs of realistic networks are typically directed to facilitate analysis.

- **State variables**: a graph’s state variables, equal in number to \( n_L \), are a set of the minimum number of variables required to describe flows within the graph. They represent the minimum number of variables an analysis must consider when solving a graph-related problem: considering more variables than the state variables is generally less efficient.

- **Subgraphs**: the subgraph of a parent graph represents a collection of nodes and links from the parent graph with the same direction and at least one reference node.

- **Spanning trees and co-trees**: a graph can be partitioned between its spanning tree (which connects all of the graph’s nodes) and co-tree (the set of chords which complements the spanning tree). Each chord added to the spanning tree forms a loop, and a graph’s number of loops is given as \( n_l = n_L - n_N + n_R \).

- **Forest and core**: similar to a tree and co-tree, a graph can be partitioned between its forest (the set of external tree branches) and its core (the set of internal looped blocks). The former represents the partition of a graph concerned with nodal balances, and the core represents the partition concerned with loop balances.

Graph theory extends well beyond the concepts presented here. For example, there are numerous applications in combinatorics and optimization, but the goal of this subsection is to provide a primer for the following sections and, ultimately, this thesis’ research. The remainder of this chapter discusses the relationship between graph theory and pipe network analyses.

3.2 Topological Relationships – Null Basis

Central to graph representation are the node-link incidence matrix \( A \in \mathbb{R}^{n_N \times n_L} \) and the loop-link incidence matrix \( B \in \mathbb{R}^{n_L \times n_L} \). They feature a distinct algebraic relationship: the column vectors of \( B^T \) span the null space of \( A \), that is, \( AB^T = 0 \). As a brief review, the null space of a matrix \( C \in \mathbb{R}^{m \times n} \) comprises the set of those vectors \( X \in \mathbb{R}^n \) that satisfy \( CX = 0 \), i.e., \( X \cap \ker(C) \), where \( \ker(C) \) denotes the *kernel or null space of* \( C \). In this sense, the collection of vectors \( X \) forms the *null basis of* \( C \).
To demonstrate that the column vectors of $B^T$ constitute the null space of $A$, consider the graph depicted in Figure 3.2b. Its node-link incidence matrix is

$$A = \begin{bmatrix}
+1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & +1 & +1 & +1 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & +1 & +1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & +1 \\
0 & 0 & 0 & 0 & -1 & -1 & 0 & +1 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\
+1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1
\end{bmatrix}$$

Our objective here is to mathematically obtain $B$ from $A$. One purely algebraic approach is to reduce $A$ to column echelon form via column operations, and the same operations are performed on an $\mathbb{R}^{n_L \times n_L}$ identity matrix. The null space vectors are then the columns of the transformed identity matrix that correspond to the zero columns in the reduced $A$. Essentially, this procedure identifies $n_l = 4$ loops by removing $n_C = 4$ chords (recall from §3.1.3 that the number of loops is $n_l = n_C = n_L - n_N + n_R$), for this leads to a set of state variables.

We begin with $A$ and an identity matrix:

$$A = \begin{bmatrix}
+1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & +1 & +1 & +1 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & +1 & +1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & +1 \\
0 & 0 & 0 & 0 & -1 & -1 & 0 & +1 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\
+1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1
\end{bmatrix}$$

First, column 3 is added to column 1, and then column 2 is subtracted from column 1 to obtain:
The first column of the reduced $A$ comprises zeros, so one of the four null space vectors has been identified (i.e., the first column of the reduced identity matrix). Next, column 6 is added to column 3, and column 5 is subtracted from column 3:

\[
\begin{bmatrix}
0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 & 0 \\
0 & −1 & 0 & 0 & +1 & +1 & 0 & 0 & 0 \\
0 & 0 & 0 & −1 & 0 & 0 & 0 & 0 & +1 \\
0 & 0 & 0 & 0 & −1 & −1 & 0 & +1 & −1 \\
0 & 0 & 0 & 0 & 0 & 0 & −1 & −1 & 0 \\
+1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
−1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
+1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\
\end{bmatrix}
\]

Another null space vector has now been obtained. Two null space vectors remain. Column 9 is now added to column 4, and column 5 is subtracted from column 4 to obtain
Lastly, columns 6 and 8 are subtracted from column 7:

Finally, for ease of reference, the columns are reordered as 2, 5, 6, 8, 9, 1, 3, 4, and 7:
Chapter 3: Pipe Network Representation

We now have a set of vectors that span the null space of $A$. They are the columns in the lower-right block of the reduced identity matrix, and with some column sign changes, the null space vectors exactly match the rows of the loop-link incidence matrix below:

$$
\begin{bmatrix}
+1 & 0 & 0 & 0 & 0 \\
0 & +1 & 0 & 0 & 0 \\
-1 & 0 & +1 & 0 & 0 \\
0 & 0 & 0 & 0 & +1 \\
0 & -1 & -1 & +1 & -1 \\
0 & 0 & 0 & -1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & +1 & 0 & 0 \\
0 & 0 & +1 & 0 & 0 \\
0 & 0 & 0 & +1 & 0 \\
0 & 0 & 0 & +1 & 0 \\
\end{bmatrix}
$$

This is no coincidence. The two match because each loop in a graph represents a closed path, and a closed path starts and ends at the same node. Furthermore, the links corresponding to the null space vectors (links L2, L5, L6, L8, and L9) represent a set of chords in the graph, while the remaining links constitute a spanning tree (Figure 3.6).

![Figure 3.6: Spanning tree and chords for null space example](image-url)
Chapter 3: Pipe Network Representation

In the case of large pipe networks, the null basis is most readily formed using concepts from graph theory, such as identifying the chords to a spanning tree, rather than through algebraic manipulation alone. This is advantageous because it requires no floating point operations. Since the graph of a network may have numerous loops, and thus numerous unique spanning trees, there are equally many null bases – some may have desirable properties (e.g., sparsity, easy factorization) while others do not. Altogether, identifying the null basis of a network’s node-link incidence matrix is necessary for loop-based solution methods.

3.3 Application to Pipe Networks

Graph theory notation provides a convenient means of describing pipe networks and key relations. Before relating it to conservation equations for mass and momentum, the data and notation needed to describe the hydraulics of arbitrary networks is reviewed first.

3.3.1 Network Properties

Just as graphs comprise node and link elements, so too do models of pipe networks. Data are required to describe the properties of nodes and links, as well as operating controls and boundary conditions. This information is often readily available from design specifications, operating protocols, process narratives, and GIS data. Though not a part of the underlying mathematical modeling framework, it helps to review the data required to describe pipe networks. The reader is also referred to Haestad Methods et al. (2003) and Thorley (2004), who provide detailed treatments of the information necessary for unsteady flow modeling.

From a modeling perspective, pipe network data can be classified as being either static or dynamic. Static data do not change over time, such as pipe lengths and junction elevations; they also include performance curves for pumps and valve closure curves. Static data can be further classified based on whether they describe nodes (junctions, reservoirs, and tanks) or links (pipes, pumps, and valves). In addition to start and end nodes, static link data include the following:

- **Pipes**: lengths, diameters, roughness, local loss coefficients, and wave speeds;
- **Pumps**: characteristic curves, and rotational inertia (efficiency curves, brake horsepower curves, and motor efficiencies are only required for energy analyses); and
- **Valves**: characteristic curves, types (e.g., flow control valve, check valve, pressure reducing valve, pressure sustaining valve), and set points (for control valves).

Valves are often incorporated into the model representation of pipes, for example, using an
equivalent pipe length. Static data for node elements include:

- **Junctions**: elevations, demand-dependent outflows, and pressure-dependent outflows;
- **Reservoirs**: heads, head patterns; and
- **Tanks**: elevations, diameters, type (e.g., open tank or pressurized vessel), area-elevation rating curves, minimum and maximum water levels, and initial head.

Often combination air valves, surge anticipation valves, and surge relief valves are treated as nodes. Some software (e.g., Bentley’s HAMMER) additionally represent pumps and valves as nodes due of how these boundary conditions are handled numerically.

Dynamic data are more pertinent to transient flow analyses. They describe how a system’s boundary conditions evolve over time, such as demands and operations. Operating controls dictate how a network operates (e.g., when a pump is turned on and off), as well as the details of specific operations (e.g., valve closure periods). The initial steady-state conditions of an unsteady flow analysis can be considered dynamic data, for they depend on the initial boundary conditions.

### 3.3.2 Network Topology

Incidence matrices were introduced to describe the node-link and loop-link relationships of arbitrary graphs. Here they are used in a similar manner to describe the topology of arbitrary pipe networks, which vary in size and complexity. Graphs comprise interconnected nodes and links, and pipe networks, having nodes (junction, reservoir, and tank elements) and links (pipe, pump, and valve elements), are treated similarly.

Consider the system below (Figure 3.7; Onizuka, 1986). It comprises $n_N = 10$ nodes (6 junctions, 3 reservoirs, and 1 tank) and $n_L = 11$ links (9 pipes and 2 valves). From §3.1, a graph must have at least one reference node: for pipe networks, tanks and reservoirs serve this purpose. The network below is connected because each node can be connected to each other node by a path, and its graph (Figure 3.7b) is directed since each link has a direction. Link directions may be arbitrary for pipes – this simply introduces a sign convention whereby positive flows are oriented in the same direction as the pipe. Link direction is, however, important for pumps and valves, a topic discussed in the following section.
Inspection of the network shows that there are too few loops to describe all link flows. To overcome this, the pseudo links \{L12, L13, L14\} are introduced (Figure 3.8): these fictitious links allow the set of loops \{l1, l2, l3, l4, l5\} to describe all of the network’s flows. The node-link-loop relationship can also be observed. For the original graph (with \(n_N = 10\) and \(n_L = 11\) links), there are \(n_l = n_L - n_N + 1 = 2\) loops (shown as loops \(l_4\) and \(l_5\) in Figure 3.8), whereas the modified graph (with \(n_N = 10\) and \(n_L = 14\)) has \(n_l = n_L - n_N + 1 = 14 - 10 + 1 = 5\) loops.

---

**Figure 3.7:** Sample pipe network schematic and graph (Onizuka, 1986)

**Figure 3.8:** Sample pipe network graph with pseudo-links and loops (Onizuka, 1986)
The node-link incidence matrix for the modified pipe network is

\[
A = \begin{bmatrix}
+1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\
-1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & +1 & 0 & 0 & -1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1
\end{bmatrix}
\]

and the loop-link incidence matrix is

\[
B = \begin{bmatrix}
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
+1 & 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & +1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & +1 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

From §3.1, the state variables of a graph are a set of the minimum number of variables required to describe the graph’s link flows. Both \( A \) and \( B \) are capable of describing the flows through the network (Figure 3.8), but \( A \) contains some redundant rows. Because \( B \) describes the network’s loops, and since the number of loops is equivalent to the number of state variables, \( B \) comprises a set of the minimum number of variables required to describe the flows in the network. The only drawbacks to \( B \) are that, numerically, it requires the identification of a null basis of \( A \), and pseudo-links must be added.

### 3.3.3 Boundary Conditions

Boundary conditions represent physical constraints that are imposed on a pipe network’s hydraulics. The dynamic data discussed in §3.3.1 are in fact boundary conditions: each prescribed demand, head-discharge relationship, and change in operating condition is a boundary condition. Although there are numerous types of boundary conditions, only common elements (reservoirs, junctions, tanks, pipes, pumps, and valves) are considered here for simplicity.

In describing the boundary conditions of pipe networks, the following notation is introduced:

- \( F(Q) \in \mathbb{R}^{n_L} \) Link head loss vector (m), where \( F(*) \) is a head loss operator
- \( H \in \mathbb{R}^{n_N} \) Nodal head vector (m)
- \( h \in \mathbb{R}^{n_L} \) Head difference vector (m)
Nodal boundary conditions are categorized according to their characteristics. Shimada (1989) considered three types of nodes: those with unknown head and known outflow (junctions with fixed demands); nodes with unknown head and unknown outflow junctions with pressure-dependent outflows and tanks); and nodes with known head and unknown outflow (reservoirs). Others have opted to only distinguish between nodes with known and unknown head (Todini and Pilati, 1988; Todini, 2011; Giustolisi et al., 2012; Todini and Rossman, 2013). The latter convention is followed here. Nodes with known head (reservoirs) are referred to as type 0 nodes, and nodes with unknown head (junctions and tanks) are classified as type 1 nodes. This allows junctions (with demand-driven, pressure-dependent, or mixed outflows) and tanks to be grouped together; the only difference between the two is that the latter has non-zero storage.

Given the above, the nodal head vector, nodal outflow vector, and node-link incidence matrix are each partitioned as follows:

\[
H^T = \begin{bmatrix} H_0^T & H_1^T \end{bmatrix} \tag{3.4}
\]
\[
q^T = \begin{bmatrix} q_0^T & q_1^T \end{bmatrix} \tag{3.5}
\]
\[
A^T = \begin{bmatrix} A_0^T & A_1^T \end{bmatrix} \tag{3.6}
\]

where \( H_0 \in \mathbb{R}^{n_0} \) and \( H_1 \in \mathbb{R}^{n_1} \) (m), \( q_0 \in \mathbb{R}^{n_0} \) and \( q_1 \in \mathbb{R}^{n_1} \) (m\(^3\)/s), \( A_0 \in \mathbb{R}^{n_0 \times n_L} \), \( A_1 \in \mathbb{R}^{n_1 \times n_L} \), and the subscripts 0 and 1 indicate the node type.

Link boundary conditions comprise pipes, pumps, and valves. As a boundary condition, pipes can be thought of as a passive element, for they dissipate hydraulic energy (through friction and local losses) with fixed properties (i.e., length, diameter, and roughness). Unlike pipes, pumps and valves are active elements. Pumps add energy (the amount of which varies with their flow according to their head curves), and they can be turned on and off; additionally, pump speeds can be adjusted to change pump performance curves. Similar to pipes, valves dissipate energy, but for purposes of providing hydraulic control. Some valves are inactive (e.g., isolation valves), though others, such as automatic control valves, constantly self-modulate their throttle position to adjust heads and flows. An example is a pressure reducing valve that limits its downstream pressures to below a prescribed pressure set point.

For incompressible flow modeling, pipes, pumps, and valves are all represented using a head loss-flow relationship. However, under compressible flow conditions, pipes are treated separately,
for their wave propagation characteristics must be tracked in order to accurately capture the resulting unsteady hydraulics. Only incompressible link representations is considered herein, though the treatment for unsteady-compressible flow follows closely.

The head loss through a pipe $j$ is expressed as

$$F(Q_j) = K_j Q_j |Q_j|^{n-1} \quad \text{(3.7)}$$

where $Q_j$ = the flow (m$^3$/s), $K_j$ = the pipe resistance (s/m$^n$), and $n$ = a head loss exponent. For the Darcy-Weisbach friction head loss model, $n = 2$, and $n = 1.852$ for the Hazen-Williams friction head loss model. The absolute value sign in Equation [3.7] implies that there is always a head loss in the direction of the flow. This is because pipes always dissipate energy: if $Q_j$ is positive, $F(Q_j)$ is also positive, and if $Q_j$ is negative, $F(Q_j)$ is also negative. The resistance term is given by

$$K_j = \begin{cases} 
\frac{8}{\pi^2} \left( \frac{L_j f_j}{D_j^5} \right) + \left( \frac{8k_j}{\pi^2 D_j^5} \right) |Q_j|^{2-n} \quad \text{(a) Darcy – Weisbach} \\
\frac{10.67 \times L_j}{D_j^{4.87} C_j} + \left( \frac{8k_j}{\pi^2 D_j^5} \right) |Q_j|^{2-n} \quad \text{(b) Hazen – Williams} 
\end{cases} \quad \text{(3.8)}$$

where $f_j$ = the Darcy-Weisbach friction factor, $C_j$ = the Hazen-Williams roughness, $D_j$ = the pipe diameter (m), $L_j$ = the pipe length (m), and $k_j$ = a local loss coefficient.

After EPANET 2 (Rossman, 2000), the head loss across a pump $j$ can be represented by

$$F(Q_j) = \begin{cases} 
-\omega_j^2 \left[ a_j - b_j \left( \frac{|Q_j|}{\omega_j} \right)^{c_j} \right] + b_j \quad \text{(a) Power curve} \\
-\omega_j^2 \left[ a_j \left( \frac{|Q_j|}{\omega_j} \right) + b_j \right] \quad \text{(b) Piecewise – linear curve} 
\end{cases} \quad \text{(3.9)}$$

where $a_j$, $b_j$, and $c_j$ = pump performance curve coefficients, and $\omega_j$ = the relative pump speed. The coefficients are constants for the power curve representation, and they are interpolation coefficients for a piecewise-linear curve. Because positive $F(Q_j)$ indicates a head loss, the negative signs in Equation [3.9] indicate a head gain since pumps provide head.

Lastly, the head loss across a valve is modeled as

$$F(Q_j) = \left( \frac{1}{\tau_j E_j} \right)^2 Q_j |Q_j| \quad \text{(3.10)}$$

where $\tau_j$ = the effective valve opening, and $E_j$ = the valve conductance (m$^{2.5}$/s). An opening of $\tau = 1$ represents a fully open valve, while $\tau = 0.25$ corresponds to a valve that is 25% open. For
control valves, $\tau_j$ is automatically selected according the valve’s set points.

### 3.4 Network Equations

The final component of this chapter connects all of the concepts presented up until now. With graph theory for representing network topology, knowledge of the network data required, and boundary conditions for imposing prescribed hydraulic constraints, expressions for mass and momentum conservation can be assembled.

#### 3.4.1 Conservation of Mass

Our first conservation law is that of mass. Essentially, the net mass entering a control volume is equal to the time rate of change in mass within the control volume: i.e., if there is a net positive inflow, the control volume’s fluid mass increases over time. A control volume can be taken about an arbitrary node, where the fluid is assumed to be incompressible at the node due to nodes being small relative to often large network sizes. Mass conservation is thus invoked as a conservation of volume. Figure 3.9 depicts flows in and out of different nodes for both types of nodes. Though both junctions and tanks are type 1 nodes, only the latter has non-zero storage.

![Diagram of nodal continuity](image)

*Figure 3.9: Nodal continuity – net inflow and outflow*

Using the incidence matrix and vector notation introduced thus far, network expressions for nodal continuity can be assembled for type 0 and type 1 nodes:

\[
A_0 Q + q_0 = 0
\]  \quad [3.11]

\[
\frac{d}{dt} (C_1 \cdot H_1) + A_1 Q + q_1 = 0
\]  \quad [3.12]

where $C_1 \in \mathbb{R}^{n_1}$ = a vector of type 1 node surface areas ($m^2$). Type 1 node outflows may be
demand-driven where \( q_1(t) \); pressure-dependent where \( q_1(H_1) \); or mixed where \( q_1(H_1, t) \). More broadly, the type 1 node outflow vector is thus \( q_1(H_1, t) \). The left-most term in Equation [3.12] represents the change in type 1 node volume with respect to time: for non-prismatic area profiles, the type 1 node surface area vector is simply \( C_1(H_1) \).

Storage is the key difference between the two types of nodes. From a modeling perspective, type 0 nodes are treated as having infinite storage, whereas type 1 nodes have finite storage. Accordingly, no area term appears in Equation [3.11] for reservoirs. If such an area term were considered, its elements would be infinitely large, effectively rendering \( \partial H_0 / \partial t = 0 \) (Giustolisi et al., 2012). Junctions and tanks have zero and non-zero finite storage. Given these definitions, the nodal conservation equations can be collectively expressed as

\[
\frac{d}{dt} (C \cdot H) + AQ + q = 0
\]  

Equation [3.13] is the network equation for the nodal mass conservation. It is valid for both compressible and incompressible flows, for the hydraulics are assumed to be incompressible at all nodes, regardless of the type of model. The left-most term represents the time rate of change of mass in all nodes. Of the terms on the left hand side, the second two represent the net flow entering a network’s nodes, where positive \( q \) indicates outflows. In developing pipe network models, only Equation [3.12] is concerned because \( H_0 \) is represented as being time-invariant.

### 3.4.2 Conservation of Momentum

In order to formulate the network equation for momentum conservation, we first require an expression that relates the link head differences and nodal heads. Using \( A \), a network’s link head differences are expressed as

\[
h = A^T H
\]  

The \( j \)-th element, \( h_j \), represents the head difference for link \( j \), which starts at node \( i \) and ends at node \( k \). To illustrate this relation, consider Equation [3.14] for the network in Figure 3.8:
Further to the above, a fundamental relationship between a network’s loops and head differences can be observed. Recall from §3.1.1 that a closed path or loop starts and ends at the same node, so the net head differences accumulated along any loop are zero for any network. Mathematically,

\[ B h = 0 \]  \hspace{2cm} [3.15]

Note that Equation [3.15] is only valid for incompressible flow conditions. The loop-head difference relationship can be observed by evaluating the terms of Equation [3.16] for the looped network from §3.4.1 (Figure 3.8):
Using a network’s incidence matrices, head difference vector, and head loss vector, a network-wide expression for momentum conservation under incompressible flow can be obtained. This relationship is fundamental to incompressible flow models: it enforces the physics of fluid resistance and inertia. Many authors in the steady-state literature refer to this as a conservation of energy. In the context of unsteady flow modeling, it can be described as one of momentum, for the relationship includes an inertial term. Further, it is obtained from the momentum equation, itself originating from Newton’s Second Law. Nonetheless, there are equivalences under many circumstances.

Under incompressible flow conditions, the momentum equation for an individual link \( j \) is

\[
\frac{L_j}{gA_j} \frac{dQ_j}{dt} + F_j - h_j = 0, \quad h_j = H_k - H_i \tag{3.16}
\]

Equation [3.16] states that the head difference vector \( h_j \) is equivalent to the sum of the inertial and fluid resistance terms. Using the node-link incidence matrix \( A \), equation [3.16] can be expressed for a network as

\[
L_I \frac{d}{dt} (Q) + F(Q) - h = 0 \tag{3.17}
\]

where \( L_I \in \mathbb{R}^{n_I \times n_L} = \) a diagonal link inertia matrix with elements of \( L_{Ij} = L_j/gA_j \) (s\(^2\)/m\(^2\)). If loops are considered, Equation [3.15] allows Equation [3.17] to be rewritten as the smaller system

\[
B \left( L_I \frac{d}{dt} (Q) + F \right) = 0 \tag{3.18}
\]

Like Equation [3.15], Equation [3.18] states that the sum of the head differences around each loop in a network is zero, but here it shows how this relates to fluid inertia and head losses.

Because only the type 1 node heads need to be solved for, it is helpful to partition the head difference term. From Equations [3.4], [3.6], and [3.11], \( h \) becomes

\[
h = A_0^T H_0 + A_1^T H_1 \tag{3.19}
\]

By combining Equations [3.17] and [3.19], the network momentum equation now becomes

\[
L_I \frac{d}{dt} (Q) + F - A_0^T H_0 - A_1^T H_1 = 0 \tag{3.20}
\]
which relates inertial effects, friction head losses, and head differences for each link in a network. Again, it is emphasized that the above are valid for incompressible flow conditions.

Together, the type 1 node mass conservation network expression (Equation [3.13]) and either of Equations [3.18] or [3.20] form the basis of incompressible network flow models. They represent a combined system of equations with \( n_L + n_1 \) unknowns (\( n_L \) unknown flows and \( n_1 \) unknown heads for type 1 nodes) and an equal number of equations. Owing to the non-linear head loss term, the system of equations cannot be solved directly; rather, numerical methods are required to obtain approximate albeit precise solutions.

### 3.5 Recent Developments

Pipe network analyses frequently require multiple simulations to evaluate various scenarios and optimize designs. It has also become common practice to consider large numbers of elements; thus, computational efficiency is an active research topic, even with significant advances in computing power. Central to recent studies are techniques from graph theory.

To improve computational efficiency, many research efforts seek to reduce the size of the network equation being solved. Deuerlein (2008) introduced a formal procedure for decomposing a network’s graph into its external forest and internal core. Since the former is represented by a set of linear equations and the latter nonlinear equations, separating the two facilitates efficient analyses by only considering the non-linear component on its own. Forest-core partitioning (see §3.1.3) permits decreases the number of variables that must be considered for demand-driven analyses, thus lowering computational run time. Simpson et al. (2014) demonstrated that forest-core partitioning can lead to significant computational savings (the authors showed savings of 40% to 70% for eight examples). With the same objective, Giustolisi et al. (2012) proposed a matrix transformation methodology that groups series nodes together. Compared to Deuerlein (2008), Giustolisi et al. (2012) neglected graph simplifications (specifically, removing chords) that would otherwise yield further reductions in the size of the problem.

Zhang et al. (2013) extended Deuerlein’s (2008) work by subdividing a network’s graph into sectors. Unlike forest-core partitioning, this approach compartmentalizes the graph into collections of cores, trees, and bridges. The study primarily concerned WDS optimization: by subdividing a network’s graph into different cores, the authors proposed optimizing the individual sectors rather than optimizing the network as a whole. Di Nardo et al. (2014) also studied pipe network optimization via network decomposition. Unlike Zhang et al. (2013), Di Nardo et al.
applied their work to two real WDSs.

In addition to eliminating serial nodes and network partitioning, null space formulations have proven promising. Standard solutions (Todini and Pilati, 1988; Todini and Rossman, 2013) require the solution of an $\mathbb{R}^{n_L+n_L+n_1 \times n_L+n_1}$ system of equations, whereas null space formulations resolve an $\mathbb{R}^{n_L+n_1 \times n_L+n_1}$ system of equations (recall that typically $n_L \ll n_1$). Elhay et al. (2014) presented the reformulated co-tree flows method, an improvement upon the co-tree method (CTM; Rahal, 1995). The CTM requires the identification of a spanning tree and co-tree: the latter is used to identify loops, develop the loop-link incidence matrix, and thus describe a system using a set of state variables. Elhay et al. (2014) reported computational savings between 15% and 82% for eight pipe networks, a considerable amount. Abraham and Stoianov (2015) also employed a null space formulation while testing other developments. In the same paper, the authors also provide detailed discussions of saddle point problems and their relation to the predominant steady-state solver (Todini and Pilati, 1988).

Rather than focus on reducing the size of the network equation, some studies have focused on those computations that require greater effort. Solving the large linear system of equations has proven to be the most computationally intensive operation. Zecchin et al. (2012) used the algebraic multigrid (AMG) method to improve this: the AMG method involves resolving sets of small approximations of the system. In addition to the linear solver step, re-computing link resistances and head losses represents another computational burden. Because most flows converge within the first few iterations, Abraham and Stoianov (2015) proposed a partial update algorithm: for this, only those resistance coefficients, head losses, and flows are updated for non-convergent links at each iteration. This alone yielded savings of 10%.

Central to the aforementioned studies are steady flow analyses. Analyses of unsteady pipe network flows using concepts from graph theory are less common. Of course, the topic thus far has only concerned the analysis of pipe network hydraulics, and there is an even broader literature base for computational fluid dynamics (CFD), which covers multi-dimensional fluid flow problems and more detailed phenomena. It is beyond the scope of this work to review the CFD literature, though the reader should know that it concerns similar topics, namely, the numerical solution of similarly structured systems of non-linear equations, some techniques of which use concepts from graph theory. For further reading, the reader is directed to Benzi et al. (2005) who review the existing literature that largely pertains to fluid flow modeling.
3.6 Concluding Remarks

Fundamental concepts from graph theory were introduced in this chapter. After briefly reviewing the data required for typical unsteady flow simulation, pipe network representation was discussed, and graph theory was shown to provide a simple yet comprehensive representation of topological relations and boundary conditions. Mathematical descriptions of common elements were reviewed, namely those for tanks, junctions, reservoirs, pipes, pumps, and valves. Using incidence matrix and vector notation, network expressions for the conservation of nodal mass and link momentum were formulated. These represent the core of incompressible flow models, and they are a key component of unsteady-compressible flow models.

Although beyond the scope of this work, the solution of large sparse systems of equations is both relevant and important to this work. The reader is referred to Strang (1986), as well as Benzi et al. (2005) for more detailed discussions. Much of this chapter is referenced from the former, while Benzi et al. concern saddle point problems, a class of algebraic equations which encompasses the types of solution expressions found in pressurized pipe network formulations. The following chapters continues from here by presenting novel modeling formulations. The material presented in this chapter is fundamental to the network representation used throughout the remainder of this thesis.
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

While surveying the literature, it was found that existing RWC formulations exhibit numerical instabilities and computational inefficiencies. Moreover, there have been no recent efforts to address these; to overcome such implementation challenges, this chapter presents two novel RWC formulations that have improved efficiency and stability. The work in this section is based on a manuscript prepared by the author and the author’s supervisor, Professor Bryan Karney. The article, entitled “Improved Rigid Water Column formulation for Simulating Slow Transients and Controlled Operations,” was published electronically in the Journal of Hydraulic Engineering on 2016 May 09.

4.1 Abstract

Rigid water column (RWC) models simulate the unsteady-incompressible hydraulics of pressurized pipe networks. They conceptually lie between water hammer and quasi-steady models, yet despite their intrinsic strengths, existing RWC formulations suffer efficiency-, stability-, and interpretation-related challenges; thus, they are overlooked as an alternative. To address the aforementioned limitations, this article presents the RWC global gradient algorithm (GGA), a novel formulation for pipe networks that has greater efficiency and overcomes the numerical challenges. The RWC GGA extends the generalized GGA (G-GGA) to consider inertial effects in addition to variable-area tanks and mixed (i.e., demand and pressure-dependent) outflows. Two pipe networks of simple and moderate complexity are used to compare the new approach against two other RWC algorithms, the G-GGA, and a water hammer model: the current work is shown to have improved stability and efficiency relative to previous work. The RWC GGA is also found to have a computational cost only slightly greater than that of the G-GGA for the same time-step size. Overall, this work highlights the practical utility of RWC models to simulate slow transient events and controlled operations.

4.2 Introduction

The occurrence of transient conditions in pipe networks is unavoidable. Depending on the forcing, hydraulic transitions can be slow with gradual changes, or they can be abrupt with potentially dramatic or detrimental consequences. Because multiple representations exist,
modeling inevitably involves selecting an appropriate formulation while addressing the trade-off between efficiency and physical accuracy. Water hammer models consider compressibility and inertial effects, so they are suited to simulating fast transients in which a network’s structural integrity is of primary concern. Comparatively, quasi-steady models simulate slow changes by presuming negligible dynamic effects; thus, they are suitable for extended period simulations in which hydraulic capacity is of interest. The inertially-based RWC models span the relatively neglected middle ground. They simulate those intermediate transient conditions that are fast enough to evoke inertial effects yet too slow to require the compressibility considerations of water hammer models.

Because RWC models use a larger time step than water hammer models and do not require the spatial discretization of pipes, they are more efficient when compressibility effects are negligible. Unlike quasi-steady models, RWC models physically account for inertial effects; moreover, the range of validity of quasi-steady models is poorly known (and often neglected by practitioners), and its results may seem valid even when the basic assumption of slow changes has been invalidated. Despite these intrinsic advantages, RWC models are seldom used due to implementation challenges and the misconception that they have little utility.

In comparing transient models, Cabrera et al. (1995) suggest that the lack of interest in RWC models is due to the perception that data uncertainty has a greater effect on hydraulics than accounting for inertial effects. Most practitioners also consider transient analyses to be readily addressed by water hammer models; however, such models still require significantly greater computational effort than RWC models, a reality that is especially important when multiple simulations are performed. Steady-state models require much less computational effort than transient simulations, but even improving their efficiency remains an active research topic (e.g., Elhay and Simpson, 2011; Giustolisi et al., 2012b; Deuerlein et al., 2015).

This article addresses the aforementioned issues by presenting the RWC global gradient algorithm (GGA). This novel formulation has improved stability and efficiency over previous work, and it closely resembles the generalized GGA (G-GGA; Giustolisi et al., 2012a), which itself is an improvement upon Todini (2011) and an extension of the well-known GGA used by EPANET 2 (Todini and Pilati, 1988; Rossman 2000). Shimada’s (1989) incidence method, another RWC algorithm, is also coupled with implicit numerical integration to form the implicit incidence method (IIM): this overcomes the incidence method’s stiffness-related challenges. The IIM serves to both validate the RWC GGA and demonstrate its performance.
4.3 Background – Unsteady Flow Modeling

Many topics touch on the current work. These include quasi-steady models, RWC formulations, and the application of inertial models to pressurized pipe networks. Steady-state modeling has received much attention in the literature; because detailed discussions can be found elsewhere (e.g., Ormsbee, 2006; Todini and Rossman, 2013), the topic is excluded here. Following the brief review below, the governing equations of unsteady-compressible flow are discussed, particularly in how they relate to the different types of models.

4.3.1 Recent Developments in the Literature

As an extension to steady-state models, so-called quasi-steady models simulate slow varying flow conditions. The term quasi-steady herein refers to an approximation when flow conditions vary with time (and are thus unsteady in a strict sense), but when inertial and compressibility effects are negligible compared with other terms; thus, it is reasonable to make the approximation $\frac{\partial Q}{\partial t} \approx 0$, thereby neglecting the acceleration term in the momentum equation. Rao and Bree (1977), Rao et al., (1977), and Bhave (1988) were the first to develop formal quasi-steady models: the authors coupled steady-state formulations with the explicit Euler and modified Euler methods to integrate the mass conservation equation for tanks. In this way, the steady-state network equation and tank continuity equation were decoupled and solved separately. To overcome numerical oscillations when multiple tanks are separated by small head losses, Todini (2011) introduced a quasi-steady variant of the original GGA (Todini and, Pilati, 1988) by coupling the two sets of equations, effectively forming a network-wide implicit solution. Giustolisi et al. (2012a) later extended the work of Todini (2011) to the G-GGA by considering pressure-dependent outflows.

Similar to the various steady-state formulations (Todini and Rossman, 2013), several RWC formulations have been developed. Holloway (1985) and Onizuka (1986) each introduced loop-based RWC formulations for pipe networks: the former linearized the head loss term to obtain a semi-implicit solution, whereas the latter used explicit numerical integration to solve the unsteady-incompressible momentum equation for pipe network loops. In contrast, Shimada (1989) developed the incidence method, a hybrid formulation, by using incidence matrices to represent network topology. Like Onizuka (1986), Shimada (1989) also explicitly solved a system of first order ordinary differential equations (ODEs). The explicit incidence method (EIM) was shown to be superior to the loop-based formulations, for pseudo loops are not
required and changes in network topology (i.e., caused by link closure and opening) are easily accommodated; however, numerical instabilities became problematic. These arise due to using explicit methods to integrate a system of stiff ODES: specifically, the equations become stiff due to the prevalence of quasi-steady conditions, hydraulically short links, and closing links. To overcome this, Shimada (1989, 1992) suggested using higher-order explicit methods, a variable time step, and state-space representation, yet despite these measures, instabilities have persisted.

As an extension to Holloway (1985), Ahmed (1997) developed an RWC formulation by applying the Newton-Raphson method to the unsteady-incompressible network equation. Although shown to perform well, realistic pipe networks were not considered, and a linearized head loss term sacrificed its integrity. Moreover, variable temporal discretization was neglected (and is shown here to affect stability), and the author used time steps on the order of minutes: this skews inertial effects, sometimes leading to inaccurate simulation results.

There are few published applications of RWC models to pressurized pipe networks. Islam and Chaudhry (1998) combined a water quality model with a loop-based RWC formulation similar to that of Holloway (1985), and its simulation results differed little from those of a quasi-steady model; however, the authors only considered demands varying over the course of hours for a simple pipe network, which do little to invoke inertial effects. Axworthy (1997) developed a comprehensive transient model with a dedicated solver for each transient flow regime. Using the EIM, Axworthy demonstrated how RWC models can accurately simulate slow transients, yet the author and later Axworthy and Karney (2000) re-iterated the occurrence of instabilities that arise when using the EIM.

Previous RWC formulations clearly suffer implementation challenges. Not surprising, these models have fallen into neglect, and there have been no recent efforts to improve. To address these previously identified issues, this article demonstrates the RWC GGA, a formulation with improved stability and computational efficiency. The range of validity of RWC models remains an open question, but such work is beyond the scope of this paper.

### 4.3.2 Governing Equations and the Transient Flow Regimes

Below the governing equations of unsteady-compressible flow are revisited to illustrate how the different types of models are related. Karney (1990) and Abreu et al. (1999) provide a similar discussion, yet because a modest period of time has elapsed since their work, this section re-emphasizes the relationships between often isolated fields of steady and unsteady flow modeling.
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

Neglecting the advective terms and unsteady friction, the governing continuity and momentum equations are

$$\frac{a^2 \partial Q}{gA \partial x} + \frac{\partial H}{\partial t} = 0$$  \[4.1\]

and

$$\frac{1}{gA} \frac{\partial Q}{\partial t} + \frac{\partial H}{\partial x} + J_{QF} = 0$$  \[4.2\]

respectively, where $Q = Q(x,t)$ is the flow rate ($m^3/s$), $H = H(x,t)$ is head (m), $A =$ pipe area ($m^2$), $D =$ pipe diameter (m), $a =$ wave speed ($m/s$), $J_{QF} =$ quasi-steady unit head loss, $\rho =$ fluid density ($kg/m^3$), $g =$ acceleration due to gravity ($m/s^2$), $x =$ distance (m), and $t =$ time (s).

Equation [4.1] incorporates the key effects of fluid compressibility and conduit elasticity, and Equation [4.2] represents the effects of inertia and viscous resistance. Derivation of the governing equations can be found in standard references (e.g., Wylie and Streeter, 1978; Chaudhry, 2014).

Water hammer models simulate unsteady-compressible flow by solving Equations [4.1] and [4.2]. For unsteady-*incompressible* flow, hydraulic changes occur slowly enough such that disturbances propagate at a much smaller time scale than the changes in a network’s boundaries, leading to moderate (or even insignificant) compressibility effects; RWC models are applicable here. Under such conditions, spatial changes in flow become negligible (i.e., $\partial Q/\partial x \approx 0$), so the terms in Equation [1] tend to zero. This leaves only Equation [4.2], which reduces can be integrated along a pipe’s length to obtain

$$\frac{L}{gA} \frac{dQ}{dt} + F(Q) - H_1 + H_2 = 0$$  \[4.3\]

where $L =$ the pipe length (m); $Q = Q(t)$ is the flow ($m^3/s$); $H_1 = H_1(t)$ and $H_2 = H_2(t)$ are heads at the start and end nodes of a pipe (m), respectively; and $F(Q) =$ head loss (m). Equation [4.3] is the momentum equation for unsteady-incompressible flow. The left-most term, the acceleration head, represents inertial effects, and it is the key difference between RWC and quasi-steady models.

As the degree of unsteadiness decreases further, inertial effects become less significant. Equation [4.3] then reduces to

$$F(Q) - H_1 + H_2 = 0$$  \[4.4\]

Equation [4.4] is the steady momentum equation. Quasi-steady flow conditions are essentially
steady, but they have very slow changes (typically at the boundaries); thus, they are simulated by solving Equation [4.4]. Such models are well-suited to simulating long-term network hydraulics, such as those in water distribution systems (WDSs) over the course of days.

4.4 Incompressible Modeling Formulations

The mathematics of the RWC GGA are presented below after briefly reviewing pipe network representation. Although other authors provide comprehensive descriptions of GGA-related formulations (e.g., Todini and Rossman 2013), to which the G-GGA is an extension, the latter is presented in this paper to contrast with the RWC GGA. The IIM is also developed by applying the Newton-Raphson to an implicit formulation of Shimada’s (1989) work.

4.4.1 Network Representation and Boundary Conditions

An arbitrary pipe network can be conveniently represented as a graph comprising nodes (e.g., junctions, reservoirs, tanks) and links (e.g., pipes, pumps, valves) (Kesavan and Chandrashekar, 1972; Nahavandi and Catanzaro, 1973). Pipe network topology can be described using the node-link incidence matrix $A \in \mathbb{R}^{n_N \times n_L}$ with entries

$$A_{ij} = \begin{cases} +S_j & \text{if link } j \text{ starts at node } i \\ 0 & \text{if link } j \text{ is not incident to node } i \\ -S_j & \text{if link } j \text{ ends at node } i \end{cases}$$

where $S_j = \text{the status of link } j$ given by

$$S_j = \begin{cases} 1 & \text{if link } j \text{ is open} \\ 0 & \text{if link } j \text{ is closed} \end{cases}$$

Nodes are classified depending on their boundary conditions. Those with known heads (e.g., reservoirs) are classified as Type 1 nodes, whereas those with unknown heads are type 1 nodes. Similar to Giustolisi et al. (2012a), the latter are further classified as type 1a or 1b depending on whether they have zero (e.g., junctions) or nonzero storage (e.g., tanks), respectively. From this, the nodal head vector, nodal outflow vector, and incidence matrix can each be partitioned into two corresponding components

$$H^T = [H_0^T \ H_1^T]$$

$$q^T = [q_0^T \ q_1^T]$$

$$A^T = [A_0^T \ A_1^T]$$
where $H \in \mathbb{R}^{nN}$ = the nodal head vector (m); $q \in \mathbb{R}^{nN}$ = the nodal outflow vector (m$^3$/s); and the subscripts represent the respective nodal boundary conditions. Note that $H$ represents the total hydraulic head. Using Equations [4.7] and [4.9], the head difference vector $h \in \mathbb{R}^{nL}$ (m) is

$$h = A^T H_0 + A^T_1 H_1$$  \[4.10\]

Nodal conservation for type 1 nodes is given by

$$\frac{d}{dt}(C_1 \cdot H_1) = -A_1 Q - q_1$$ \[4.11\]

where $Q \in \mathbb{R}^{nL}$ = a link flow vector (m$^3$/s), and $C_1 \in \mathbb{R}^{n1}$ = a vector of type 1 node surface areas (m$^2$). The nodal outflow term $q_1$ represents both demand-drive and pressure-dependent outflows. The first $n_j$ elements of $C_1$ are zero for junctions, and the last $n_T$ elements comprise the tank surface areas, which may vary with head.

### 4.4.2 Generalized Global Gradient Algorithm

The G-GGA is reviewed next to contrast with the RWC GGA and IIM. Quasi-steady algorithms such as the G-GGA presume negligible compressibility and inertial effects, so they are well suited to simulating slow flow variations. The presentation below differs slightly from that of Giustolisi et al. (2012a) in the following ways: tank heads are treated as an unknown rather than the changes in tank heads over a time step; the notation is adjusted to match that of the RWC GGA (for comparison purposes); and head losses are generalized as a single term rather than using a flow coefficient. Additionally and unlike Todini (2011), the current work considers the effect of variable-area tanks on the tank head solution iterate for the sake of completeness, although this is likely only important for those tanks with highly variable areas.

The G-GGA uses the Newton-Raphson method to solve the quasi-steady network equation. Like the original GGA (Todini, and Pilati 1988), initial estimates that satisfy nodal continuity are not required, and the formulation has favorable convergence properties for demand-driven problems (Todini and Rossman, 2013). The global network equation is

$$\begin{bmatrix} F(*) \\ -A_1 \end{bmatrix} - A^T_1 [Q^{(t+1)}] - [A^T_0 H_0 b_1] = 0$$ \[4.12\]

$$G_1(*) = q_1(*) + \frac{1}{\varnothing \Delta t} C_1^{(t+1)}.$$ 

$$b_1 = \left(\frac{1-\varnothing}{\varnothing}\right)(A_1 Q^{(t)} + q_1^{(t)}) - \frac{1}{\varnothing \Delta t} C_1^{(t)} \cdot H_1^{(t)}$$
where \( F(\ast) : \mathbb{R}^{n_L} \rightarrow \mathbb{R}^{n_L} \) = a vector operator (m); \( q_1(\ast) : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_1} \) = a vector operator (m\(^3\)/s); \( G_1(\ast) : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_1} \) = a vector operator (m\(^3\)/s); \( \Delta t = \) the time step (s); \( \vartheta = \) a temporal integration parameter for type 1 nodes; \( b_1 \in \mathbb{R}^{n_1} \) (m\(^3\)/s); and \( t+1 \) indicates the solution at time step \( t+1 \).

Unlike traditional treatments, \( F(\ast) \) provides a broader representation of head losses.

The rows of Equation [4.12] represent the steady flow momentum equation (Equation [4.4]) and nodal continuity for type 1 nodes. From the initial conditions, the junction elements of \( A_2 Q(t) + q_1(t) \) are zero; thus, junction continuity at time step \( t+1 \) is not affected by \( \vartheta \) (junction elements of \( A_2 Q^{(t+1)} + q_1^{(t+1)} \) will also have zero elements). A value of \( \vartheta = 1 \) yields first-order fully implicit temporal integration for type 1 nodes, whereas \( \vartheta = 0.5 \) yields second-order central spatial weighting. Todini (2011) showed that the selection of \( \vartheta \) is important to ensure numerical stability when multiple tanks are located in close proximity with small head losses between them.

Within a quasi-steady simulation using the G-GGA, Equation [4.12] must be solved at the end of each time step. To accomplish this, the Newton-Raphson method is used to obtain a solution by minimizing residual differences, and the following iterative solution is obtained:

\[
\begin{bmatrix}
D^{(m)} & -A_1^T \\
-A_1 & -g_1^{(m)}
\end{bmatrix}
\begin{bmatrix}
Q^{(m+1)} \\
H_1^{(m+1)}
\end{bmatrix} =
\begin{bmatrix}
b_0^{(m)} \\
b_1^{(m)}
\end{bmatrix}
\]

\[ b_0^{(m)} = D^{(m)} Q^{(m)} - F^{(m)} + A_0^T H_0 \quad b_1^{(m)} = G_1(H_1^{(m)}) - g_1^{(m)} H_1^{(m)} + b_1^{(t)} \]

\[ g_{1,ii}^{(m)} = \frac{\partial d_i^{(m)}}{\partial h_i^{(m)}} + \frac{1}{\vartheta \Delta t} (c_i^{(m)} + H_i^{(m)} \frac{\partial c_i^{(m)}}{\partial h_i^{(m)}}) \]

where \( D^{(m)} \in \mathbb{R}^{n_L \times n_L} \) = the Jacobian of \( F(Q^{(m)}) \) (s/m\(^2\)) with respect to \( Q^{(m)} \); \( g_1^{(m)} \in \mathbb{R}^{n_1 \times n_1} \) = the Jacobian of \( G_1(H_1^{(m)}) \) (m\(^2\)/s) with respect to \( H_1^{(m)} \); \( m+1 \) = the current iterate to the solution at time step \( t+1 \); \( b_0^{(m)} \in \mathbb{R}^{n_L} \) (m); and \( b_1^{(m)} \in \mathbb{R}^{n_1} \) (m\(^3\)/s). The first term on the right hand side of \( g_{1,ii}^{(m)} \) above arises due to pressure-dependent outflows, while the second term accounts for the effect of variable-area tanks. Of these, the latter was omitted by Todini (2011). The values of \( D^{(m)} \) depend on the types of links (e.g., pipes, pumps, valves) and their respective characteristics: expressions for the values of \( D^{(m)} \) are summarized in the following section.

Equation [4.13] can be solved as shown above or by expanding and simplifying its rows. The latter approach, known as Schur complement reduction, yields the solution.
\[ H_1^{(m+1)} = -\left( U^{(m)} \right)^{-1} \left( b_1^{(m)} + A_1 (D^{(m)})^{-1} b_0^{(m)} \right) \]  

\[ Q^{(m+1)} = (D^{(m)})^{-1} \left( A_1^T H_1^{(m)} + b_0^{(m)} \right) \]

\[ U^{(m)} = A_1 (D^{(m)})^{-1} A_1^T + g_1^{(m)} \]

where \( U^{(m)} \in \mathbb{R}^{n_1 \times n_1} \) = the Schur complement. Because \( U^{(m)} \) is symmetric positive definite, methods such as the conjugate gradient method can be employed to solve the system of equations.

### 4.4.3 Rigid Water Column Global Gradient Algorithm

The RWC GGA represents an extension of the G-GGA (Giustolisi et al., 2012a) that incorporates inertial effects. Like the G-GGA, the RWC GGA can be used to model pressure-dependent outflows and variable-area tanks, and it is derived by applying the Newton-Raphson method to an implicit discretization of the unsteady-incompressible momentum network equation. Accordingly, a similar formulation is obtained.

Using Equation [4.10] for the head difference vector, the unsteady-incompressible momentum equation (Equation [4.3]) for a network is

\[ L_I \frac{d}{dt} (Q) + F(Q) - A_0^T H_0 - A_1^T H_1 = 0 \]  

[4.15]

where \( L_I \in \mathbb{R}^{n_L \times n_L} \) = a diagonal link inertia matrix \((s^2/m^2)\) with entries of \( L_{I,jj} = L_j / g A_j \ (s^2/m^2)\).

By using a variable implicit approximation of \( dQ/dt \), Equation [4.15] becomes

\[ \frac{1}{\theta \Delta t} L_I Q^{(t+1)} + F(Q^{(t+1)}) - A_0^T H_0 - A_1^T H_1^{(t+1)} - R^{(t)} = 0 \]  

[4.16]

\[ R^{(t)} = \frac{1}{\theta \Delta t} L_I Q^{(t)} - \left( \frac{1-\theta}{\theta} \right) \left( F(Q^{(t)}) - A_0^T H_0 - A_1^T H_1^{(t)} \right) \]

where \( \theta \) = a temporal discretization parameter for the momentum equation, and \( R^{(t)} \in \mathbb{R}^{n_L} \) = a vector \((m)\). A value of \( \theta = 1 \) yields the first order implicit backward Euler method, and \( \theta = 0.5 \) yields second order central spatial weighting.

After Todini (2011), the nodal continuity equation for type 1 nodes (Equation [4.11]) can be solved using a similar variable temporal discretization:

\[ -A_1 Q^{(t+1)} - G_1 \left( H_1^{(t+1)} \right) - b_1^{(t)} = 0 \]  

[4.17]

The unsteady-incompressible momentum (Equation [4.15]) and type 1 node continuity equation (Equation [4.17]) can be jointly expressed as
Equation [4.18] is the network equation for unsteady-incompressible flow, which closely resembles that for quasi-steady flow (Equation [4.12]).

For unsteady-incompressible simulations, Equation [4.18] is solved at the end of every time step. Similar to the G-GGA, the Newton-Raphson method can be applied to determine the solution in an iterative manner via successive approximations. This yields

\[
\begin{bmatrix}
\frac{1}{\vartheta \Delta t} L_1 + F(*) \\
-A_1 \\
-G_1(*)
\end{bmatrix} \begin{bmatrix}
Q^{(t+1)} \\
H_1^{(t+1)}
\end{bmatrix} - \begin{bmatrix}
A_1^T H_0 + R^{(t)} \\
b_1^{(t)}
\end{bmatrix} = 0
\]

[4.18]

Equation [4.18] is the network equation for unsteady-incompressible flow, which closely resembles that for quasi-steady flow (Equation [4.12]).

For unsteady-incompressible simulations, Equation [4.18] is solved at the end of every time step. Similar to the G-GGA, the Newton-Raphson method can be applied to determine the solution in an iterative manner via successive approximations. This yields

\[
\begin{bmatrix}
M^{(m)} \\
-A_1 \\
-G^{(m)}_1
\end{bmatrix} \begin{bmatrix}
Q^{(m+1)} \\
H_1^{(m+1)}
\end{bmatrix} = \begin{bmatrix}
B_0^{(m)} + R^{(t)} \\
b_1^{(m)}
\end{bmatrix}
\]

[4.19]

\[
M^{(m)} = \frac{1}{\vartheta \Delta t} L_1 + D^{(m)}
\]

where \( M^{(m)} \in \mathbb{R}^{n_L \times n_L} \) = the Jacobian of the upper-left block of Equation [4.18] (s/m²) with respect to \( Q^{(m)} \). Equation [4.19] closely resembles Equation [4.13] for the G-GGA.

For a given time step, a solution to Equation [4.18] is obtained when iterate unknowns from Equation [4.19] converge. Like Equation [4.13], Equation [4.19] has a saddle point structure. The solution iterates can thus be obtained via Schur complement reduction:

\[
H_1^{(m+1)} = -(V^{(m)})^{-1} \left( b_1^{(m)} + A_1 (M^{(m)})^{-1} (b_0^{(m)} + R^{(t)}) \right)
\]

[4.20]

\[
Q^{(m+1)} = (M^{(m)})^{-1} \left( A_1^T H_1^{(m+1)} + b_0^{(t)} + R^{(t)} \right)
\]

\[
V^{(m)} = A_1 (M^{(m)})^{-1} A_1^T + g^{(m)}_1
\]

where \( V^{(m)} \in \mathbb{R}^{n_1 \times n_1} \) = the Schur complement to Equation [4.19]. Like Equation [4.14] and \( U^{(m)} \), \( V^{(m)} \) is symmetric positive definite, so similar methods can be used to solve Equation [4.20].

### 4.4.4 Jacobian Submatrix Terms

Within the Newton-Raphson method, a Jacobian matrix is used to solve a system of nonlinear equations by minimizing residual differences via successive approximations. Values of the Jacobian submatrix \( D^{(m)} \) are calculated by evaluating \( \partial F / \partial Q \) for each link. Three link elements are considered: pipes, pumps, and general valves. Although the expressions here can be found elsewhere (e.g., Rossman, 2000; Todini, 2011), the following provides a convenient summary.

The head loss through a pipe can be expressed as:
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

\[ F(Q) = K|Q|^{n-1}Q \]  \[4.21\]

where \( F(Q) \) = the head loss (m), \( K \) = the pipe resistance (s\(^n\)/m\(^{3n-1}\)), and \( n \) = a head loss exponent (\( n = 2 \) for the Darcy-Weisbach friction head loss model, and \( n = 1.852 \) for the Hazen-Williams friction head loss model). The resistance term is

\[
K = \begin{cases} \frac{8}{g\pi^2} \frac{L_f}{D^5} + \frac{8k}{g^n D^4} |Q|^{2-n} & \text{(a) Darcy – Weisbach} \\ \frac{10.67L}{D^{4.87}C^n} + \frac{8k}{g^n D^4} |Q|^{2-n} & \text{(b) Hazen – Williams} \end{cases}
\]  \[4.22\]

where \( f \) = the Darcy-Weisbach friction factor, \( C \) = the Hazen-Williams roughness factor, and \( k \) = a local loss coefficient. Evaluating the Jacobian matrix value for a pipe \( j \) gives

\[ D^{(m)} = nK|Q|^{n-1} + (n - 2) \frac{8k}{g^n D^4} |Q| \]  \[4.23\]

Equation [4.27] neglects the flow-dependent nature of the Darcy-Weisbach friction factor, which otherwise involves evaluating the non-linear term \( \partial K/\partial Q \) (this was considered by Simpson and Elhay, 2011). The higher-order relationship is omitted here for simplicity.

Following the convention of EPANET 2 (Rossman, 2000), the head loss for a pump is:

\[
F(Q) = \begin{cases} -\omega^2 \left( a - b \left( \frac{|Q|}{\omega} \right)^c \right) & \text{(a) Power curve} \\ -\omega^2 \left( a \left( \frac{|Q|}{\omega} \right) + b \right) & \text{(b) Piecewise – linear curve} \end{cases}
\]  \[4.24\]

where \( a, b, \) and \( c \) = pump curve coefficients, and \( \omega \) = the relative pump speed. The coefficients are constants for power curves, and they are interpolation coefficients for piecewise-linear curves. Because pumps provide head, there is a negative sign. The Jacobian matrix value for a pump \( j \) is

\[
D^{(m)} = \begin{cases} \omega^{2-c}bc|Q|^{c-1}\text{sign}(Q) & \text{(a) Power curve} \\ -\omega^2a \cdot \text{sign}(Q) & \text{(b) Piecewise – linear curve} \end{cases}
\]  \[4.25\]

A valve \( j \)'s head is represented as

\[ F(Q) = \frac{1}{\tau^2 E^2} |Q|Q \]  \[4.26\]

where \( E \) = the valve conductance (m\(^{2.5}\)/s), and \( \tau \) = the effective valve opening. Values of \( \tau = 1 \) and \( \tau = 0.5 \) represent 100\% and 50\% open valves, respectively. The Jacobian value for a valve is

\[ D^{(m)} = \frac{2}{\tau^2 E^2} |Q| \]  \[4.27\]
4.4.5 Some Remarks on the G-GGA and RWC GGA

There are a number of similarities between the G-GGA and the RWC GGA. The G-GGA solves the quasi-steady network equation (Equation [4.12]) using Equation [4.13], whereas the RWC GGA solves the unsteady-incompressible network equation (Equation [4.19]) using Equation [4.20]. Because the latter follows a similar solution derivation to the former with only a few additional terms (specifically, $M^{(m)}$ and $R^{(t)}$), the RWC GGA solution closely resembles that for the G-GGA. As a result, the RWC GGA can benefit from GGA-related work in the literature, such as simplifying distributed demands along pipes (Giustolisi and Todini, 2009; Berardi et al., 2011; Giustolisi and Laucelli, 2011), implementing solution techniques for pressure-dependent outflows (Ang and Jowitt, 2006; Giustolisi et al., 2008), and reducing the dimension of the solution equations to improve efficiency (Giustolisi et al., 2012b; Deuerlein et al., 2015). However, extending some of these techniques to the RWC GGA will require adjustments.

The main advantage of the G-GGA is that its stability does not depend on $\Delta t$ (given an appropriate value for $\theta$; Todini 2011). This is because $\Delta t$ only affects nodal conservation for tanks, which are often few in number; thus, large $\Delta t$ can be used to achieve greater efficiency. However, the Jacobian submatrix $D^{(m)}$ may become singular when zero flows arise, and convergence is inhibited by pressure-dependent outflows. Accordingly, appropriate approaches are required to resolve singularity issues (e.g., Elhay and Simpson, 2011) and ensure convergence (e.g., Ang and Jowitt, 2006, introduced artificial reservoirs; Giustolisi et al., 2008, implemented step-size control; and Elhay et al., 2015, approached the problem as one of optimization with a damping scheme). For the RWC GGA, $\Delta t$ also influences how inertial effects are handled. If $\Delta t$ is too large, inertial effects will be smeared, and they will be poorly modeled. As a result, large $\Delta t$ are indeed permissible, but accurate simulation using the RWC GGA requires $\Delta t$ on the order of seconds (when inertial effects are present).

Because the RWC GGA uses a variable implicit solution to the unsteady-incompressible momentum equation, a key issue occurs when links are opened and closed. When a network’s topology changes (e.g., a pump is turned on or a valve is closed), a fully implicit solution (i.e., $\theta = 1$) helps to maintain nodal continuity. When a link is closed, the corresponding value of $R^{(t)}$ should be set to zero to ensure zero flow in the link at the time of closure; accordingly, residual flows need not be allocated across a pipe network as in Axworthy and Karney (2000). The RWC GGA does not encounter numerical challenges when zero flows arise, for the Jacobian submatrix...
\( M^{(m)} \) is nonsingular as a result of the presence of the non-zero inertial term. Thus, special measures are not needed for zero flows, provided that a suitable \( \Delta t \) is used.

Physically, \( M^{(m)} \) and \( R^{(t)} \) of the RWC GGA represent the discretized inertial terms from the unsteady-incompressible momentum equation (Equations [4.3] and [4.16]). The acceleration head characterizes the magnitude of inertial effects: in the absence thereof, the inertial terms cancel out, and the RWC GGA effectively reduces to the G-GGA. An alternative interpretation of this (albeit one that is numerical) is that inertial effects are characterized by the nonzero component of the vector on the right side of Equation [4.12], which is

\[
\begin{bmatrix}
F(*) & -A_1^T \\
-A_1 & -G_1(*)
\end{bmatrix}
\begin{bmatrix}
Q^{(t+1)} \\
H_1^{(t+1)}
\end{bmatrix}
- \begin{bmatrix}
A_0^T H_0 \\
b_1^{(t)}
\end{bmatrix}
= \begin{bmatrix}
-L_I \frac{d}{dt} (Q) \\
O_{n_1}
\end{bmatrix} \neq 0
\]

where \( O_{n_1} \in \mathbb{R}^{n_1} \) = a zero vector. By implementing a quasi-steady algorithm such as the G-GGA when inertial effects are present, the acceleration head term is interpreted as a residual to be reduced to zero: in this sense, quasi-steady models artificially accelerate a network’s hydraulics toward equilibrium. If inertial effects are present and considered, fluid inertia will resist flow changes and prolong the time required to reach equilibrium.

### 4.4.6 Implicit Incidence Method

The incidence method (Shimada 1989) involves numerically integrating a system of first-order ODEs that represent the unsteady-incompressible network equation. Shimada (1989, 1992) and others (Axworthy, 1997; Axworthy and Karney, 2000) noted that when used with explicit integration methods, the EIM often requires extremely small time steps or complex variable time step methods to overcome stiffness-related challenges. To supplement the shortcomings of explicit methods, the IIM presented here uses the Newton-Raphson method to solve an implicit formulation of the incidence method’s system of ODEs.

Neglecting pressure-dependent outflows, the system of ODEs of the incidence method is

\[
\frac{d}{dt} (X) = \begin{bmatrix}
-W \cdot F(*) & WA_1^T H_0 - T \\
-C_1b A_1b & O_{n_r \times n_r}
\end{bmatrix} \begin{bmatrix}
Q \\
H_1b
\end{bmatrix} + \begin{bmatrix}
WA_1^T H_0 - T \\
O_{n_r}
\end{bmatrix}
\]

where \( X \in \mathbb{R}^{n_L + n_T} \) = a vector of unknown link flows and tank heads (\( m^3/s \mid m \)); \( C_{1b} \in \mathbb{R}^{n_T \times n_T} = \)}
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

a diagonal matrix of tank areas (m²); \( O_{nT \times nT} \in \mathbb{R}^{nT \times nT} = \) a zero matrix; \( O_{nT} \in \mathbb{R}^{nT} = \) a zero vector; \( R \in \mathbb{R}^{nL \times nL} = \) a matrix (m²/s²), \( S \in \mathbb{R}^{nL \times nL} = \) a matrix, and \( W \in \mathbb{R}^{nL \times nL} = \) a matrix (s²/m²); \( T \in \mathbb{R}^{nL} = \) a vector (m³/s²); and \( I_{nL \times nL} \in \mathbb{R}^{nL \times nL} = \) an identity matrix. Note that the subscripts \( 1a \) and \( 1b \) respectively indicate junctions and tanks. The first row of Equation [4.28] represents the unsteady-incompressible momentum equation and junction continuity, and the second row represents nodal conservation for tanks.

Using a finite difference treatment of \( dX/dt \), the system of equations can be rearranged to

\[
\frac{d}{dt} \begin{pmatrix} X(t+1) \\ \end{pmatrix} - \frac{1}{\Delta t} X(t+1) + \left( \frac{1-\theta}{\theta} \right) \frac{d}{dt} \begin{pmatrix} X(t) \\ \end{pmatrix} + \frac{1}{\theta \Delta t} X(t) = 0 \tag{4.29}
\]

Applying the Newton-Raphson method to Equation [4.29] leads to the iterative solution

\[
\begin{pmatrix} WD^{(m)} + W A^{T}_{tb} C_{1b}^{-1} A_{1b} & -W A^{T}_{tb} C_{1b}^{-1} A_{1b} \\ O_{22} & I_{nLT} \\ \end{pmatrix} \begin{pmatrix} Q^{(m+1)} \\ H^{(m+1)} \\ \end{pmatrix} = \begin{pmatrix} W b^{(m)} - T \\ O_{nT} \\ \end{pmatrix} + \left( \frac{1-\theta}{\theta} \right) \frac{d}{dt} \begin{pmatrix} X(t) \\ \end{pmatrix} \tag{4.30}
\]

where \( I_{nL+nT} \in \mathbb{R}^{nL+nT \times nL+nT} = \) an identity matrix.

A solution to Equation [4.29] is obtained when the successive iterates calculated using Equation [4.30] converge. Like the RWC GGA, the IIM maintains nodal continuity when links are opened and closed if \( \theta = 1 \) and if the corresponding value on the right hand side of Equation [4.30] is set to zero. Once the link flows and tank heads at the end of a time step have been solved for, the junction head vector is calculated as

\[
H^{(t+1)}_{1a} = R^{-1} \left( A_{1a} L_{1a}^{-1} \left( F^{(t+1)} - A^{T}_{1b} H^{(t+1)}_{1b} - A^{T}_{0} H_{0} \right) - \frac{d}{dt} \left( q_{1a}^{(t+1)} \right) \right) \tag{4.31}
\]

To summarize the IIM, an implicit formulation of the incidence method’s system of ODEs (Equation [4.28]) is solved using Equation [4.30]. Unlike Shimada’s (1989) EIM, the IIM uses implicit numerical integration to avoid the stiffness-related challenges faced by the latter.

### 4.5 Performance of the RWC Formulations

Two pipe networks demonstrate the current work. The first simple system validates the RWC GGA and IIM, while the second provides a more practical test. Note that Ahmed’s (1997) work is excluded because of its similarity to the RWC GGA, and because the latter is a more robust formulation. Quasi-steady models are indeed ill-suited to simulating flow variations on the order of seconds, yet the comparisons illustrate the presence and magnitude of inertial effects. Run times are reported only for informative purposes. The water hammer simulations used wave speeds of
1,000 m/s, and the comparison of results permits the evaluation of the importance of compressibility effects: this permits the evaluation of compressibility effects. A value of \( \theta = 0.5 \) was used for the RWC GGA and G-GGA. Finally, the EIM was implemented using the modified Euler method; although higher-order methods can be used, they would undoubtedly still encounter instabilities. This section concludes with a discussion of computational efficiency and stability.

### 4.5.1 Example 1 – Small Pipe Network

Pipe network 1 (Figure 4.1; Onizuka 1986), a small fictitious network, is used to validate the current work. The system is initially at steady state with both valves fully open, each having a conductance of 0.87 \( m^{2.5} / s \). Reservoirs R1, R2, and R3 have constant heads of 50, 0, and 0 m, respectively, and the tank has a diameter of 10 m and initial head of 37.96 m. The system’s link properties and initial flows are provided below (Table 4.1).

![Figure 4.1: Pipe network 1 (Onizuka, 1986)](image)

<table>
<thead>
<tr>
<th>Link</th>
<th>( D ) (mm)</th>
<th>( L ) (m)</th>
<th>( \varepsilon ) (mm)</th>
<th>( Q_0 ) (m(^3)/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1,000</td>
<td>1,000</td>
<td>0.65</td>
<td>1.79</td>
</tr>
<tr>
<td>P2</td>
<td>300</td>
<td>1,000</td>
<td>0.95</td>
<td>0.22</td>
</tr>
<tr>
<td>P3</td>
<td>500</td>
<td>2,000</td>
<td>0.60</td>
<td>0.05</td>
</tr>
<tr>
<td>P4</td>
<td>1,000</td>
<td>2,000</td>
<td>0.63</td>
<td>1.57</td>
</tr>
<tr>
<td>P5</td>
<td>500</td>
<td>50</td>
<td>0.85</td>
<td>0.00</td>
</tr>
<tr>
<td>P6</td>
<td>500</td>
<td>1,000</td>
<td>0.85</td>
<td>0.78</td>
</tr>
<tr>
<td>P7</td>
<td>500</td>
<td>1,000</td>
<td>0.85</td>
<td>0.79</td>
</tr>
<tr>
<td>P8</td>
<td>500</td>
<td>1</td>
<td>0.85</td>
<td>0.96</td>
</tr>
<tr>
<td>P9</td>
<td>500</td>
<td>1</td>
<td>0.85</td>
<td>0.84</td>
</tr>
<tr>
<td>V1</td>
<td>500</td>
<td>–</td>
<td>–</td>
<td>0.96</td>
</tr>
<tr>
<td>V2</td>
<td>500</td>
<td>–</td>
<td>–</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table 4.1: Dimensions and initial flows for pipe network 1
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

Transient conditions are induced by the simultaneous partial uniform closure of both valves by 90% over 100 s. Results from the models are shown below for a 500 s simulation (Figure 4.2). Using $\Delta t = 1$ s and $\theta = 0.5$, the RWC models produce identical results, and a comparison against the G-GGA results (also based on $\Delta t = 1$ s) illustrates how inertial effects limit the time rate of change, thereby increasing heads and prolonging the time until equilibrium. There is little difference between the RWC and water hammer results, which are based on $\Delta t = 0.01$ s: this suggests the absence of compressibility effects, so the RWC models are valid here. Figure 4.2 also shows that the network has not reached the new steady state after 500 s, but it instead requires approximately 4,000 s as a result of the relatively large size of tank T1.

Although the RWC models’ simulation results (Figure 4.2) are comparable, there are significant computational differences (Table 4.2). The RWC GGA and IIM have run times that are approximately 20 and 130% greater than that of the G-GGA, respectively, whereas the run time of the EIM is greater than that of the G-GGA by over 500%.

<table>
<thead>
<tr>
<th>Table 4.2: Formulation run times relative to those of the G-GGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formulation</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>G-GGA</td>
</tr>
<tr>
<td>RWC GGA</td>
</tr>
<tr>
<td>IIM</td>
</tr>
<tr>
<td>EIM</td>
</tr>
<tr>
<td>Water Hammer (MOC)</td>
</tr>
</tbody>
</table>

The simulation results illustrate how each model simulates partial valve closure. However, the predicted hydraulics differ little between the RWC and quasi-steady models: this is because of the relatively large tank T1, which has a moderate cushioning effect. To illustrate the effect of changing the tank size, additional simulations were performed for pipe network 1 with a tank diameter of 1 m (Figure 4.3). A comparison of Figure 4.3 against Figure 4.2 shows that decreasing the tank size largely affects the transient hydraulics: there are now noticeable oscillations with the reduced tank diameter. There remains little difference between the results of the water hammer and RWC models, so changing the tank size only modestly affects the validity of the latter. The G-GGA also fails to simulate the oscillatory inertial hydraulics: a key dynamic that would otherwise be overlooked.
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

Figure 4.2: Simulation results for pipe network 1 with a tank diameter of 10 m

(a) Flow traces for pipes P5 and P8

(b) Head traces at junction J2 and tank T1
Figure 4.3: Simulation results for pipe network 1 with a tank diameter of 1 m
4.5.2 Example 2 – Skeletonized Water Distribution System

The second pipe network analyzed (Figure 4.4) is a skeletonized WDS that represents a pressure zone within a larger regional system. The WDS model consists of 34 junctions, 3 reservoirs, 44 pipes, and 2 pump stations each with 4 parallel pumps. The system supplies a total demand of 0.51 m$^3$/s, and excess flow entering the WDS leaves through reservoir R3 for another pressure zone. The system is initially at steady state with two small pumps on at pump station PS1 and one large pump on at PS2.

![Pipe network 2 – water distribution system](image)

Figure 4.4: Pipe network 2 – water distribution system

A common operation for WDSs is turning pumps on and off. Here, a third medium-sized pump at pump station PS1 is turned on and then off as a controlled operation (Figure 4.5). The pump start operation involves turning the pump on against a closed discharge valve and opening the valve over a period of 25 s, and the pump stop operation involves the exact opposite. The valve closure curve is represented as a cubic curve. A value of $\theta = 0.8$ was selected with $\Delta t = 1$ s to maintain stability, as greater values for $\theta$ produced unstable results.

Figure 4.5 shows that the RWC GGA and IIM produce identical results. As expected, the G-GGA failed to simulate any inertial effects, the most important of which being the downsurge at 125 s. The EIM encountered numerical instabilities, even for a time step of $\Delta t = 0.001$ s, so its results are not shown. Similar to pipe network 1, the simulation results for the water hammer for ($\Delta t = 0.01$ s) and RWC models agree closely with the slight exception of the downsurge at 125 s, where compressibility effects are indeed present. Although the RWC models simulate a greater downsurge than the water hammer model, their validity is acceptable for practical purposes. This comparison also illustrates how compressibility effects reduce the magnitude of the downsurge in the system. Finally, unlike pipe network 1, the WDS reaches its new steady states quickly as a result of an absence of storage.
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

Figure 4.5: Simulation results for pipe network 2

(a) Pump 3 and pump station flow traces

(b) Pump station discharge head traces
A comparison of the models’ run times (Table 4.2) shows that the RWC GGA has a run time that is only 11% greater than that of the G-GGA for the same time-step size, a rather modest amount. Comparatively, the IIM has a run time that is 140% greater than that of the G-GGA. As mentioned previously, the EIM was notably unstable, so no results are shown.

### 4.5.3 Efficiency and Stability

From the preceding examples, the RWC GGA features greater stability and efficiency than the RWC alternatives. It is difficult to firmly state how much more (or less efficient) one formulation is compared to another, for this strongly depends on the topic network. To empirically investigate how $\Delta t$ and $\theta$ affect the run time, additional simulations were performed for pipe network 1 with a tank diameter of 1 m (Table 4.3). The results illustrate the run times of the current RWC formulations relative to those of the G-GGA for various $\Delta t$ and $\theta$. Table 4.3 shows that there is greater variation due to $\Delta t$ than $\theta$, and greater values for $\theta$ marginally reduce the run time. Additionally, the RWC GGA is consistently more efficient than the IIM.

**Table 4.3: RWC GGA and IIM pipe network 1 run times relative to those of the G-GGA**

<table>
<thead>
<tr>
<th>$\Delta t$ (s)</th>
<th>RWC GGA</th>
<th>IIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\theta = 1$</td>
<td>$\theta = 0.75$</td>
</tr>
<tr>
<td>0.25</td>
<td>1.18</td>
<td>1.17</td>
</tr>
<tr>
<td>0.5</td>
<td>1.18</td>
<td>1.17</td>
</tr>
<tr>
<td>1</td>
<td>1.23</td>
<td>1.22</td>
</tr>
<tr>
<td>2</td>
<td>1.28</td>
<td>1.27</td>
</tr>
<tr>
<td>2.5</td>
<td>1.18</td>
<td>1.16</td>
</tr>
<tr>
<td>5</td>
<td>1.24</td>
<td>1.23</td>
</tr>
</tbody>
</table>

Although efficiency depends largely on $\Delta t$, stability is strongly influenced by $\theta$. For the RWC GGA, inspection of Equations [4.17] and [4.20] shows that $\theta = 0$ results in dividing by zero; thus, the algorithm is unstable for $\theta$ close to zero. Regarding the IIM, as $\theta$ approaches zero, its behavior begins to mimic that of the EIM, which has been shown both in this study and elsewhere (e.g., Axworthy, 1997; Axworthy and Karney, 2000) to be unsuitable for large networks and large $\Delta t$. Additionally, simulation results for pipe network 2 showed that the appropriate selection of $\theta$ also depends on the size of a network and its transient hydraulics.
4.6 Discussion

Overall, the present RWC formulations performed well relative to the EIM. Both the RWC GGA and IIM produced stable and similar results, but the former was found to be more efficient. Similar to the different steady-state formulations (Todini and Rossman 2013), this is likely the result of matrix symmetry and ease of inversion: that is, the coefficient matrix of the RWC GGA (Equation [4.20]) is symmetric and thus readily solved. Comparatively, that of the IIM (Equation [4.30]) is not, so its numerical solution is slower. The IIM also requires the calculation of a greater number of intermediate matrices, further contributing to its lower efficiency.

Simulation results from pipe networks 1 and 2 showed that, not surprisingly, \( \Delta t \) strongly affects efficiency and \( \theta \) affects stability. The present formulations performed well for constant time steps up to \( \Delta t = 5 \) s (larger \( \Delta t \) are permissible, although too large of values would skew inertial effects) and were stable for \( \theta \) closer to unity. From this, \( \Delta t \) on the order of 1–2 s and \( \theta \) near unity should be sufficient for moderately complex pipe networks when inertial effects are present. As is natural for stable methods, smaller time steps eventually converge to a single transient response, albeit with greater computational costs. For highly complex pipe networks, a fully implicit solution (i.e., with \( \theta = 1 \)) may be necessary to maintain stability, but this needs to be investigated further. The practical challenge is of course to choose \( \Delta t \) and \( \theta \) that together produce a simultaneously computationally efficient, accurate, and stable solution.

In the case of both pipe networks, the quasi-steady G-GGA failed to capture the key dynamics of slow transients. For pipe network 2, inertial effects were found to be relatively short-lived in the absence of storage, so they have little effect on the network’s long-term hydraulics, thereby confirming the suitability of algorithms such as the G-GGA. Additionally, the fact that the simulation results from the water hammer and RWC models agreed closely for both pipe networks (within 3 m of head) signifies that compressibility effects were not significant; therefore, the operations were sufficiently slow and the RWC models reasonably valid. Overall, if the operations invoked significant compressibility effects, both RWC and quasi-steady models would become invalid, yet the former would be more informative of this than the latter (which was the case for pipe network 2). These examples also demonstrated the ability of RWC models to indicate whether inertial effects are present, in addition to their utility in simulating controlled operations.

4.7 Conclusions and Broader Considerations

This chapter extended the G-GGA to an RWC algorithm so that inertial effects can be
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

included in addition to mixed outflows and variable-area tanks. The IIM was also developed by coupling Shimada’s (1989) incidence method with implicit numerical integration. Compared with previous work, the current formulations better maintain nodal continuity when links are opened and closed, they have improved stability, and they permit a flexible temporal discretization. In addition to being one of the most efficient RWC formulations to date, a distinct advantage of the RWC GGA is its similarity to the G-GGA; accordingly, it can benefit from the GGA-related literature. Furthermore, the RWC GGA does not require adjustments to handle matrix singularity as a result of zero flows, an advantage over the G-GGA, nor does it require complex techniques for handling link closures. The aforementioned come at a modest increase in computational effort and necessitate the use of a time step that is suitable to analyzing slow transients.

Two pipe networks are used to demonstrate and compare the RWC formulations. The first validated the current work, and the second demonstrated the practical ability of RWC models to simulate slow transient conditions and controlled operations in a moderately complex system. The present formulations perform well, avoiding the challenges faced by the EIM, and simulation results from the G-GGA and a water hammer model were respectively used to illustrate the role of inertial effects and assess the validity of the RWC models. The RWC GGA was provisionally shown (at least in these limited test cases) to be the most efficient published RWC formulation. Future work needs to determine the range of validity of each type of model. This is particularly important for RWC models, and it is expected that they will play a key role in characterizing the transient flow regimes with their intermediate ground between the extremes of water hammer and quasi-steady models.
Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

Notation

\( \theta \) Temporal discretization factor
\( \vartheta \) Tank time integration parameter
\( \tau \) Effective valve opening for valve \( j \)
\( \omega \) Relative pump speed
\( \Delta t \) Time step size (s)
\( A \) \( \mathbb{R}^{N \times L} \) Node-link incidence matrix
\( A_0 \) \( \mathbb{R}^{n_0 \times n_L} \) Type 0 node-link incidence matrix
\( A_1 \) \( \mathbb{R}^{n_1 \times n_L} \) Type 1 node-link incidence matrix
\( A_{1b} \) \( \mathbb{R}^{n_T \times n_L} \) Tank-link incidence matrix
\( A_j \) Area of pipe \( j \) (m\(^2\))
\( a_j, b_j, c_j \) Curve coefficients for pump \( j \)
\( b_0^{(m)} \) \( \mathbb{R}^{n_L} \) Solution vector (m)
\( b_1^{(m)} \) \( \mathbb{R}^{n_1} \) Solution vector (m\(^3\)/s)
\( b_1^{(f)} \) \( \mathbb{R}^{n_1} \) Solution vector (m\(^3\)/s)
\( C_1 \) \( \mathbb{R}^{n_1 \times n_1} \) Diagonal type 1 node area matrix (m\(^2\))
\( C_{1b} \) \( \mathbb{R}^{n_T \times n_T} \) Tank surface area matrix (m\(^2\))
\( C_j \) Hazen-Williams roughness of pipe \( j \)
\( D^{(m)} \) \( \mathbb{R}^{n_L \times n_L} \) Jacobian sub-matrix (s/m\(^2\))
\( D_j \) Diameter of pipe \( j \) (m)
\( E_j \) Conductance of valve \( j \) (m\(^{2.5}\)/s)
\( F(*) \) \( \mathbb{R}^{n_L} \rightarrow \mathbb{R}^{n_L} \) Head loss vector operator (m)
\( f_j \) Darcy-Weisbach friction factor for pipe \( j \)
\( G_1(*) \) : \( \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_1} \) Mass balance vector operator (m\(^3\)/s)
\( g \) Acceleration due to gravity (m/s\(^2\))
\( H \) \( \mathbb{R}^{n_N} \) Node head vector (m)
\( H_0 \) \( \mathbb{R}^{n_0} \) Type 0 node head vector (m)
\( H_1 \) \( \mathbb{R}^{n_1} \) Type 1 node head vector (m)
\( h \) \( \mathbb{R}^{n_L} \) Head difference vector (m)
\( i \) Start node index
### Chapter 4: Improved Modeling Formulations for Unsteady-Incompressible Flow

- \( j \) Link index
- \( K_j \) Resistance for pipe \( j \) (\( s^n/m^{3n-1} \))
- \( k \) End node index
- \( k_j \) Local loss coefficient for pipe \( j \)
- \( L_I \) Diagonal link inertia matrix (\( s^2/m^2 \))
- \( L_j \) Length of pipe \( j \) (m)
- \( M^{(m)} \) Jacobian sub-matrix
- \( m + 1 \) Current iteration index
- \( n \) Head loss model exponent
- \( n_L \) Number of links
- \( n_J \) Number of junctions
- \( n_N \) Number of nodes
- \( n_R \) Number of reservoirs
- \( n_T \) Number of tanks
- \( Q \) Link flow vector (m\(^3\)/s)
- \( q \) Nodal outflow vector (m\(^3\)/s)
- \( q_0 \) Type 0 node outflow vector (m\(^3\)/s)
- \( q_1(*) \) Type 1 node outflow vector operator (m\(^3\)/s)
- \( R \) Incidence method solution matrix (m\(^3\)/s\(^2\))
- \( r^{(t)} \) Intermediate solution vector
- \( S \) Incidence method solution matrix
- \( t + 1 \) Current time step index
- \( U^{(m)} \) Intermediate solution matrix
- \( V^{(m)} \) Intermediate solution matrix
- \( W \) Incidence method solution matrix
- \( X \) Vector of unknown link flows and tank heads (m\(^3\)/s | m)
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flow

In the previous chapter, the rigid water column (RWC) global gradient algorithm (GGA) was introduced. It closely resembles the generalized global gradient algorithm (G-GGA; Giustolisi et al., 2012), with the only difference being the inertial term; otherwise, both formulations feature similar structures. To benefit from their advantages, this chapter presents an adaptive hybrid incompressible solver termed the hybrid global gradient algorithm (HGGA). This chapter is based on an article published in the Journal of Hydraulic Engineering on 2016 July 12, and it is titled “Adaptive hybrid transient formulation for simulating incompressible hydraulics.”

5.1 Abstract

Many studies have aimed to characterize pressurized transient hydraulics. However, it remains difficult to assess the importance of dynamic effects in a robust manner, and modeling is further complicated by the tension between computational efficiency and physical accuracy. To address such challenges for incompressible network flows, this chapter presents an adaptive modeling approach that combines a novel hybrid formulation, termed the hybrid global gradient algorithm (HGGA), with unsteady flow characterization indices and an adaptive scheme. The HGGA combines the generalized and rigid water column global gradient algorithms, thus enabling it can adapt to inertially-dominated flows and those without such effects. Three physically-based indicators are then introduced to characterize unsteady flow: these actively inform the HGGA of how to model a system. Together, the adaptive scheme informs the HGGA of how to model a system according to the unsteady flow indicators: essentially, computational efficiency and physical accuracy are balanced by only using smaller time step and RWC solver when necessary. Two pipe networks are used to demonstrate the current work. The first illustrates the utility of the inertial indicators, and the second comprises an extended period simulation with the adaptive scheme. While more computationally intensive than conventional modeling, the methodology provides a better representation of dynamic hydraulics.

5.2 Introduction

In the twentieth century, many authors sought to characterize transient flows to justify approximate analyses. Two notable early works are those of Joukowsky (1898) and Allievi
(1913): the fundamental Joukowsky equation relates pressure changes to abrupt flow changes, and the dimensionless Allievi parameters characterize the time rate of change of boundary conditions relative to the wave travel time. Later studies (Parmakian, 1955; Valentine, 1965; Karney, 1990; Wood et al., 1990; Cabrera et al., 1995; Axworthy, 1997; Abreu et al., 1999) investigated the boundaries between the different transient models, yet despite the modest devotion to the subject, characterizing the complex nature of unsteady flow remains challenging.

A key issue in simulating unsteady flow is model selection. Advances in computational power have permitted the analysis of large pipe networks, but model validity is still only sometimes clear. For example, so-called quasi-steady (i.e., non-inertial unsteady) models are typically used to simulate the long-term hydraulics of water distribution systems (WDSs); however, the assumption of negligible dynamic effects is often invalidated by even controlled operations (Filion and Karney, 2002). Indeed, many practitioners blindly employ transient models. Reliance on questionable data can lead to overlooking critical system states, inadequate designs, increased costs, and even system failures. At the opposite extreme, those models with greater physical accuracy cannot always be applied, for their use is accompanied by increased computational demand; it is key to balance computational efficiency with physical accuracy. Accordingly, there is a need to better understand transient pipe network hydraulics (which can be accomplished by developing insightful unsteady flow indicators) in order to discern when to apply which type of model. Addressing this will lead to more accurate and economical transient modeling practices.

One approach to resolving the accuracy-efficiency trade-off is adaptive hybrid modeling. Such models combine multiple types of transient solvers, and they actively select which to employ according to the simulated flow conditions. Axworthy (1997) and Filion and Karney (2002) are two such examples. The former developed a comprehensive hybrid model with a dedicated solver for each of the transient flow regimes, and the latter combined a water hammer model with a quasi-steady model to perform extended period simulations (EPSs) of WDSs. Because Axworthy (1997) used Shimada’s (1989) incidence method with explicit integration, the criterion for switching between the rigid water column (RWC) and non-inertial unsteady solvers was based on the stability of the incidence method. This is formulation-specific and somewhat physically arbitrary. Filion and Karney’s (2002) model successfully simulated a range of transient flows; however, it was noted to be computationally expensive, particularly for large networks. Additionally, Filion and Karney used a fixed duration for the water hammer microsimulations, so either key transient hydraulics may be overlooked or unnecessary time steps may be implemented.
This work builds on Axworthy (1997) and Filion and Karney (2002). A modeling approach is presented that combines a hybrid formulation, termed the hybrid global gradient algorithm (HGGA), with an adaptive scheme to simulate the incompressible hydraulics of pressurized pipe networks. The HGGA combines the generalized global gradient algorithm (G-GGA; Giustolisi et al., 2012), a quasi-steady unsteady solver, and the RWC global gradient algorithm (GGA; Nault and Karney, 2015; Chapter 4), both of which are improvements upon Todini’s (2011) work. To balance the competing objectives of computational efficiency and physical accuracy, the time step is adapted to a network’s unsteadiness: that is, a smaller time step is only used to track more rapid changes when greater physically accuracy is required. Three physically-based unsteady flow indicators are proposed for characterizing inertial effects. Of these, the first two are based on Karney’s (1990) integrated energy equation, and the third relates a network’s inertial effects to its heads. These indicate whether inertial effects are present and thus which solver should be used; together with the HGGA and adaptive scheme, these form an adaptive hybrid transient model (AHTM). The methodology is demonstrated using two pipe networks: the first illustrates the utility of the inertial indicators, and the second comprises an EPS of a water transmission network.

5.3 Hybrid Global Gradient Algorithm

A similar treatment to the G-GGA (Giustolisi et al., 2012) and the RWC GGA (Chapter 4; Nault and Karney, 2016) is used to form the HGGA. Though, it is based on a hybrid momentum equation. In addition to considering pressure-dependent outflows, variable area tanks, and inertial effects, the HGGA features the ability to implement either solver.

5.3.1 Hybrid Incompressible Momentum Equation

Unlike water hammer models, incompressible flow models are valid when changes at a network’s boundaries occur slowly enough relative to the wave travel time such that the fluid is approximately incompressible. The unsteady-incompressible momentum equation for a pipe is

\[
\frac{L}{gA} \frac{dQ}{dt} + F(Q) - H_1 + H_2 = 0
\]  

[5.1]

where \( F(Q) \) = the head loss (m), and \( H_1 \) and \( H_2 \) = the heads at the upstream and downstream nodes (m), respectively. The left-most term in Equation [5.1], the acceleration head, represents inertial effects: if it is small, inertial effects are insignificant, but if this term is large, inertial effects and possibly even compressibility effects are present. The terms “small” and “large” here...
are relative and rather difficult to define.

Between RWC and quasi-steady (i.e., non-inertial unsteady) models the key distinction is their treatment of the momentum equation. The former resolves Equation [5.1], whereas non-inertial models assume that flow changes occur slowly enough such that \( dQ/dt \approx 0 \) for all pipes. This results in a negligible acceleration head, leading Equation [5.1] to reduce to

\[
F(Q) - H_1 + H_2 = 0
\]  

Because the acceleration head is the only difference between Equations [5.1] and [5.2], it is central to delineating the boundary between RWC and non-inertial models. In a more general sense, Equations [5.1] and [5.2] can be combined into a hybrid equation as

\[
S_I \frac{L}{gA} \frac{dQ}{dt} + F(Q) - H_1 + H_2 = 0
\]  

where \( S_I \) is a binary parameter for whether inertial effects are modeled (\( S_I = 1 \)) or not (\( S_I = 0 \)).

### 5.3.2 Network Equation and Solution

To obtain a network expression, Equation [5.3] is rewritten using network notation, combined with an expression for nodal continuity, and then solved. The network form of Equation [5.3] is

\[
\begin{bmatrix}
\frac{1}{\theta \Delta t} S_I L_I + F(*) & -A^T_1 & \mathbf{b}^{(t)}_1 \\
-A_1 & -G_1(*) & \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
Q^{(t+1)}_1 \\
H^{(t+1)}_1 \\
0
\end{bmatrix}
- A^T_1 H_0 + S_I R^{(t)} = 0
\]  

where \( F(*) : \mathbb{R}^{n_L} \rightarrow \mathbb{R}^{n_L} \) is a vector operator (m); \( G_1(*) : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_1} \) is a vector operator (m³/s); \( L_I \in \mathbb{R}^{n_L \times n_L} \) is a diagonal link matrix with entries of \( L_{Ijj} = L_j/gA_j \) (s²/m³); \( A_0 \in \mathbb{R}^{n_0 \times n_L} \) and \( A_1 \in \mathbb{R}^{n_1 \times n_L} \) are node-link incidence matrices for type 0 nodes (e.g., reservoirs) and type 1 nodes (e.g., junctions and tanks), respectively; \( C^{(t+1)}_1 \in \mathbb{R}^{n_1 \times n_1} \) is diagonal matrix of type 1 node areas (m²); \( \Delta t \) is the time step (s); \( \theta \) is a momentum integration parameter; \( \theta \) is a nodal continuity integration parameter; \( R^{(t)} \in \mathbb{R}^{n_L} \) is an integration term for the previous time step (m); and \( \mathbf{b}^{(t)}_1 \in \mathbb{R}^{n_1} \) is an integration vector for the previous time step (m³/s). The reader is referred to Todini (2011), Giustolisi et al. (2012), and
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

Chapter 4 (also Nault and Karney, 2015) for more detailed definitions.

Equation [5.4] closely resembles those network expressions for the G-GGA and RWC GGA. The first row represents the network form of Equation [5.3], and the second row enforces nodal conservation for type 1 nodes (i.e., junctions and tanks). Physically, $S_I$ determines whether a network is modeled with ($S_I = 1$) or without ($S_I = 0$) inertial effects at a given time step: the appropriate selection $S_I$ is one subject of the current work. A value of $\theta = 1$ yields a fully implicit solution to Equation [5.1], while $\theta < 1$ yields a semi-implicit solution to Equation [5.3]. The parameter $\theta$ follows the same convention as $\theta$ but for type 1 node mass conservation.

Applying the Newton-Raphson method to Equation [5.4] leads to the iterative solution

$$
\begin{bmatrix}
M^{(m)} & -A_1^T \\
-A_1 & -g_1^{(m)}
\end{bmatrix}
\begin{bmatrix}
Q^{(m+1)} \\
H_1^{(m+1)}
\end{bmatrix}
= 
\begin{bmatrix}
b_0^{(m)} + S_I R^{(t)} \\
b_1^{(m)}
\end{bmatrix}
$$

[5.5]

where $M^{(m)} \in \mathbb{R}^{n_L \times n_L}$ is the Jacobian of the upper-left block of Equation [5.4] with respect to $Q^{(m)}$ (s/m$^2$); $D^{(m)} \in \mathbb{R}^{n_L \times n_L}$ is the Jacobian of $F(Q^{(m)})$ with respect to $Q^{(m)}$ (s/m$^2$); $g_1^{(m)} \in \mathbb{R}^{n_1 \times n_1}$ is the Jacobian of $G_1^{(m)}$ with respect to $H_1^{(m)}$ (m$^2$/s); $m+1$ = the current iterate to the solution at time step $t+1$; $b_0^{(m)} \in \mathbb{R}^{n_L}$ (m); and $b_1^{(m)} \in \mathbb{R}^{n_1}$ (m$^3$/s). A summary of expressions for the elements of $D^{(m)}$ is provided in Chapter 4.

As a saddle point problem, Equation [5.5] can be solved using Schur complement reduction:

$$
H_1^{(m+1)} = - (V^{(m)})^{-1} \left( b_1^{(m)} + A_1 (M^{(m)})^{-1} (b_0^{(m)} + S_I R^{(t)}) \right)
$$

[5.6]

$$
Q^{(m+1)} = (M^{(m)})^{-1} \left( A_1^T H_1^{(m+1)} + b_0^{(t)} + S_I R^{(t)} \right)
$$

$$
V^{(m)} = A_1 (M^{(m)})^{-1} A_1^T + g_1^{(m)}
$$

where $V^{(m)} \in \mathbb{R}^{n_1 \times n_1}$ = the Schur complement to Equation [5.5].

5.3.3 Some Remarks on the HGGA

Owing to its generality, the HGGA offers a few key modeling advantages. A comparison of the HGGA, G-GGA (Giustolisi et al., 2012), and RWC GGA solutions (Chapter 4; Nault and
Karney, 2015) shows that the only differences are $S_I$ in $M^{(m)}$ and that preceding $R^{(t)}$, so $S_I$ effectively controls whether the HGGA represents the G-GGA or RWC GGA. This allows the simulation of both inertially-dominated and quasi-steady flows. By including the inertial terms, the importance of inertial effects can be accurately assessed. Further, since a water hammer model is avoided, larger $\Delta t$ are permissible; $\Delta t$ must only be sufficiently small to resolve inertial effects when the HGGA is implemented as an RWC model. Overall, the HGGA’s wide physical modeling range is its main advantage.

Physically, $S_I$ represents whether or not inertial effects are modeled. That is, if $S_I = 1$, the HGGA becomes the RWC GGA, and if $S_I = 0$, the HGGA reduces to the G-GGA. It may be specified prior to performing a simulation, but a better approach is to provide a means for the automatic selection of $S_I$. Indicators are introduced below that serve this purpose.

### 5.4 Inertial Flow Indicators

Three transient flow indicators are presented below to characterize inertial effects. The first two are based on Karney’s (1990) integrated energy equation, and the third relates the inertial contribution of a network’s hydraulics to its junction heads. These indicators inform the HGGA of which solver to use.

#### 5.4.1 Inertial Energy Indicators

In studying transient flow, Karney (1990) suggested comparing the components of an integrated energy equation to delineate the boundaries of the unsteady flow regimes. That between unsteady-incompressible and quasi-steady flow was not emphasized, but the energy equation nonetheless finds utility here for exactly that. The integrated energy equation for unsteady-compressible flow in a pipe is (Karney, 1990)

$$\frac{dU}{dt} + \frac{dT}{dt} + D' + W' = 0$$

\[5.7\]

\[
\frac{dU}{dt} = \frac{1}{2} \rho A \left( \frac{g}{a} \right)^2 \frac{d}{dt} \int_{x=0}^{L} (H)^2 \, dx
\]

\[
\frac{dT}{dt} = \left( \frac{\rho}{A} \right) Q \frac{d}{dt} \int_{x=0}^{L} Q \, dx
\]

\[
D' = \rho g \int_{x=0}^{L} (J_{QF} + J_{UF}) \cdot Q \, dx
\]

\[
W' = \rho g (Q_2 H_2 - Q_1 H_1)
\]

where $dU/dt$ = the time rate of change in internal energy (J/s); $dT/dt$ = time rate of change in total kinetic energy (J/s); $D'$ = rate of viscous dissipation (J/s); $W'$ = rate at which work is done at the pipe’s ends to force the fluid through the pipe (J/s); $J_{QF}$ and $J_{UF}$ = quasi-steady and unsteady
unit head losses, respectively; and the subscripts 1 and 2 respectively refer to the pipe’s start and end nodes. Unlike Karney (1990), the energy equation here (Equation [5.7]) is expressed in terms of flow rather than velocity, and the head loss term has a more general representation.

For incompressible flow, $\partial Q / \partial x \approx 0$ and $H$ varies linearly along a pipe. The internal energy term $dU/dt$ thus tends to zero, leading Equation [5.7] to reduce to

$$\frac{d}{dt}(T_j) = \rho g Q_j \cdot H_{ij}$$
$$D'_j = \rho g Q_j F_j$$
$$W'_j = \rho g Q_j h_j$$

for a pipe $j$, where $H_{ij}$ is the acceleration head (m), and $h_j$ is head difference (m). Only $D'_j$ and $W'_j$ sum to zero for quasi-steady flow models, for such formulations neglect inertial effects.

Inertial effects are represented by $dT'_j/dt$. Similar to Karney’s (1990) compressibility index, the relative importance of inertial effects can be assessed by comparing $dT'_j/dt$ to the other terms in the energy equation. A comparison of $dT'/dt$ against $W'_j$ shows that their quotient is simply that of the acceleration head and the head difference. An important concept here is one of scales: if the acceleration head represents a large portion of a pipe’s head difference, then inertial effects are relatively important. From this, absolute and relative indicators

$$\phi_1 = \max(|H_I|)$$
$$\phi_2 = \max\left(\frac{|H_{ij}|}{|H_{ij}|+|F_j|}\right), \forall j$$

are proposed, where $\phi_1$ has units of [m], and $\phi_2$ is dimensionless. The absolute inertial energy indicator represents the maximum magnitude of inertial effects within a network, and the relative inertial energy indicator represents the relative importance of inertial effects. For example, inertial effects are present if $\phi_1$ is large, and inertial effects are important if $\phi_2$ is large. If either indicator is consistently small throughout a simulation, then inertial effects can presumably be ignored without significant penalty. Both inertial energy indicators would seem to be best used together, for the use of only one can be misleading. For example, $\phi_1$ neglects the accumulation of inertial effects along long pipelines, and $\phi_2$ is biased for pipes with extreme head losses; these shortcomings are overcome if both indicators are used simultaneously.

### 5.4.2 Inertial Junction Head

The inertial junction head (IJH) presented below provides a different physical perspective by
quantifying the contribution of each pipe’s inertia to a network’s junction heads. Consider the
network form of the unsteady-incompressible momentum equation:

$$H_I + F(Q) - A_0^T H_0 - A_{1a}^T H_{1a} - A_{1b}^T H_{1b} = 0$$ \[5.11\]

where the subscripts 1a and 1b refer to junctions and tanks, respectively. Equation \[5.11\] can be
rearranged for the junction head vector as

$$H_{1a} = (A_{1a} A_{1a}^T)^{-1} A_{1a} (H_I + F(Q) - A_0^T H_0 - A_{1b}^T H_{1b})$$ \[5.12\]

Equation \[5.12\] has four components that respectively represent the contributions of a
network’s fluid inertia, head losses, tanks, and reservoirs to the network’s junction heads. Each
term is dynamic and interrelated, though the reservoir component only changes if a network’s
topology changes (i.e., by opening and closing links). Here, the inertial component is of particular
interest. It relates system-wide inertial effects to a network’s junction heads through the incidence
matrices. While insightful, the definition above permits the possibility of positive and negative
acceleration heads cancelling out; this may leave near-zero values that otherwise suggest the
absence of inertial effects. Accordingly, the IJH indicator is proposed as

$$\phi_3 = \max \left( \left| (A_{1a} A_{1a}^T)^{-1} A_{1a} \right| \cdot |H_I| \right)$$ \[5.13\]

where $\phi_3$ has units of [m]. This indicator represents the extent to which network-wide inertial
dynamics affect a system’s junction heads, and greater $\phi_3$ indicate a greater impact thereof.

### 5.4.3 Utility and Use with the HGGA

Two key aspects of the indicators are their physical meaning and their use with the HGGA.
The indicators $\phi_1$ and $\phi_2$ respectively represent the presence and importance of inertial effects:
that is, if $\phi_1$ is large and $\phi_2$ is small, inertial effects are present yet unimportant, but if $\phi_1$ is small
and $\phi_2$ is large, inertial effects are important yet not of large magnitude. Similarly, the IJH
indicator $\phi_3$ is a measure of how a network’s unsteadiness relates to its junction heads.

When combined with the HGGA, the inertial indicators actively characterize unsteady flow
and aid the appropriate selection of $S_I$. This is accomplished by specifying tolerances beyond
which inertial effects are deemed important and thus simulated:

$$S_I = \begin{cases} 
1 & \text{if } [\phi_1 \geq \phi_1' \text{ and } \phi_2 \geq \phi_2'] \text{ or } \phi_3 \geq \phi_3' \\
0 & \text{if } [\phi_1 < \phi_1' \text{ or } \phi_2 < \phi_2'] \text{ and } \phi_3 < \phi_3' 
\end{cases}$$ \[5.14\]
where $\phi_1' = \text{the absolute inertial energy tolerance (m)}$, $\phi_2' = \text{the relative inertial energy tolerance}$, and $\phi_3' = \text{the IJH tolerance (m)}$. Though $S_I$ is considered a scalar variable here, it can be treated as a matrix with diagonal elements selected for individual pipes; however, because $\Delta t$ is consistent throughout a network, it is simpler to keep $S_I$ as a scalar parameter. When inertial effects are both present and important (that is, when $\phi_1 \geq \phi_1'$ and $\phi_2 \geq \phi_2'$), they should be simulated; otherwise they can be neglected. Key to this are the tolerances. The tolerances may initially appear arbitrary, but their flexibility is in fact an advantage. If greater physical accuracy is desired, lower tolerances can be adopted. Consequently, the adaptive modeling formulation can be tailored to provide physically accurate results or represent the objective of an analysis.

### 5.5 Adaptive Scheme and Implementation

Relating the HGGA and unsteady flow indicators is the final component to this work, the adaptive scheme. With the HGGA’s adaptive ability and the inertial indicators for characterizing unsteady flows, the adaptive scheme permits efficient simulations by adjusting the time step according to the simulated flow conditions. The scheme uses small $\Delta t$ when greater physical accuracy is required; otherwise, a larger time step is used when inertial effects are negligible.

With conventional EPSs using a quasi-steady model, transient hydraulics resulting from changes at a pipe network’s boundaries are neglected. To avoid overlooking these hydraulics, two time scales are considered similar to Filion and Karney (2002): the larger extended period time scale (on the order of minutes) and the smaller inertial time scale (on the order of seconds). The flow indicators characterize a network’s unsteadiness, and if inertial effects are deemed present, $S_I = 1$ and a series of microsimulations are performed using small $\Delta t$. Once inertial effects have dissipated, the solution resumes using larger $\Delta t$ with $S_I = 0$. Figure 5.1 illustrates this approach, and pseudo-code of the implementation is provided on the following page (Figure 5.2).

![Figure 5.1: HGGA variable time-stepping scheme](image)
1. **Front End Simulation:** At beginning of extended period \( t \) at time \( T_i \) with an event at time \( T_{i,1} \), perform a front end simulation:

\[
\begin{align*}
  t & = 0 \\
  \text{MODE} & = \text{‘quasi-steady’} \\
  \Delta t & = T_{i,1} - T_i - \Delta t_{WH} \quad \text{and} \quad T_{i,t} = T_i + \Delta t
\end{align*}
\]

Calculate the solution at time \( T_{i,t} \)

2. **Midway Microsimulations:**

\[
\begin{align*}
  \text{MODE} & = \text{‘RWC’} \\
  \Delta t & = \Delta t_{RWC}
\end{align*}
\]

While true:

\[
\begin{align*}
  t & = t + 1 \quad \text{and} \quad T_{i,t} = T_{i,t-1} + \Delta t \\
  \text{Calculate the solution at time } T_{i,t} \\
  \text{If } T_{i,t} \geq T_{i,1} + \Delta T_{op}, \{ \max \phi_1 < \phi'_1 \text{ or } \max \phi_2 < \phi'_2 \}, \quad \text{and} \quad \phi_3 < \phi'_3, \text{ end loop}
\end{align*}
\]

3. **Back End Simulation:**

\[
\begin{align*}
  \text{MODE} & = \text{‘quasi-steady’} \\
  \Delta t & = T_{i+1} - T_{i,t} \\
  \text{Calculate the solution at time } T_{i+1}
\end{align*}
\]

![Figure 5.2: Adaptive hybrid incompressible flow model implementation](image)

### 5.6 Simulation of Incompressible Flow

Two pipe networks demonstrate the methodology. The first, a pumping pipeline, illustrates the utility of the inertial indicators, and the second pipe network, a WDS, provides a more practical test. Simulation results from the HGGA are compared against those of the non-inertial G-GGA. Results from a water hammer model are excluded from the second example, for its application for an EPS is impractical. Avesani et al. (2012) suggest using \( \vartheta \) near unity to avoid numerical oscillations amongst tanks. Similarly, Nault and Karney (2016) found that \( \vartheta \) close to unity is required for a stable solution for large networks, so all simulations used \( \vartheta = \vartheta = 1 \) and \( \vartheta = 1 \).

#### 5.6.1 Example 1 – Pumping Pipeline

Figure 5.3 shows a pumping pipeline with an open surge tank. With the valve fully open and the pump on, the system initially operates at a flow of 0.62 m\(^3\)/s. Unsteady conditions are then induced by a controlled pump stop: this involves uniformly closing the valve over a period of 100 s, and then the pump is shut off. The results (Figure 5.4) show little difference between simulation data from the water hammer (using a wave speed of 1,000 m/s) and RWC models (based on \( \Delta t = 1 \) s), affirming the absence of compressibility effects. As expected, the G-GGA (also based on \( \Delta t = 1 \) s) failed to simulate the inertial effects and thus tank oscillations: although its application here is hardly appropriate, the comparison illustrates the magnitude of inertial effects.
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

Profiles of the inertial indicators are provided in Figure 5.5 and Figure 5.6 to analyze the system’s unsteady hydraulics. The solid curves in Figure 5.5a and 5.4b represent (but are not equal to) the absolute and relative inertial energy indicators, respectively, and the solid curve in Figure 5a represents the IJH indicator. From Figure 5.5a, the pipeline’s acceleration head reaches a minimum just after the valve is closed. After this, it dominates the head loss, as can also be seen in Figure 5.5b. The head component traces in Figure 5.6 illustrate how the system’s characteristics contribute to junction J2’s head. When the valve becomes fully closed at 200 s, there is an immediate and discontinuous change in the head contributions from the reservoirs and tank.

Given the simulation results above, some adjustments are needed to avoid solution discontinuities with the HGGA. The solution is of an oscillatory nature due to the surge tank, so the HGGA may abruptly switch the inertial solver (i.e., from $S_I = 1$ to $S_I = 0$) midway through the first cycle prior to inertial effects dissipating: this is because the indicators simultaneously equal zero upon flow reversal (i.e., when $dQ/dt = 0$). One means of overcoming this is to force a number of subsequent solutions using $S_I = 1$ after inertial effects have dissipated. Additionally, only $\phi'_1$ and $\phi'_3$ decay as inertial effects dissipate, so it would appear that $\phi'_2$ is best used to identify the beginning of an unsteady flow period.

With the above numerical adjustments, simulations were performed using the HGGA with three sets of tolerances (Figure 5.7). From Equation [5.14], the HGGA will only simulate those inertial effects beyond the tolerances; thus, greater values for $\phi'_1$, $\phi'_2$, and $\phi'_3$ yield a solution that more closely resembles that of a quasi-steady model. In Figure 5.7, the first two sets of tolerances show similar results, with the first set replicating those of an RWC model, but the third set only captures half of the first oscillation. Accordingly, the tolerances should reflect the purpose of an analysis. For example, lower tolerances should be adopted if inertial oscillations, such as here, are of interest, otherwise larger values can be used.
(a) Flow traces for the pipeline and tank inflow

(b) Head traces for the tank

Figure 5.4: Simulation results for pipe network 1
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

Figure 5.5: Inertial energy characteristics for pipe network 1
Figure 5.6: Head components for the pump discharge junction, pipe network 1
Figure 5.7: HGGA simulation results for pipe network 1
5.6.2 Example 2 – EPS of a Transmission Network

Here a more practical application is demonstrated. A 24 hour EPS was performed for the water network below (Figure 5.8). The system represents the transmission skeleton of a pressure zone within a regional WDS, and it comprises five primary pumping facilities, two booster pumping stations, and an elevated tower. In addition supplying reservoirs of higher elevation pressure zones, the system distributes an average flow of 2.94 m$^3$/s to local distribution systems (the demand pattern is shown in Figure 5.9). Each of Zone 1 pump stations 1 and 2, which have higher capacity, is equipped with a surge tank for controlling transient pressures. The model comprises 932 links (including 26 pumps and 159 valves) and 884 junctions.

![Pipe network 2 – water transmission network](image)

Throughout the EPS, controlled operations are performed to ensure adequate system pressures and supply zone 2’s reservoirs. The pump start operations involve starting a pump against a closed discharge valve, and then the valve is opened over 30 s using a cubic valve curve. Conversely, the pump stop operations involve closing a pump’s discharge valve over 60 s and then shutting the pump off. The initial conditions (Table 5.1) and operations log (Table 5.2) describe the system conditions.
Simulation results from the present AHTM and a conventional model (using the non-inertial G-GGA with a fixed $\Delta t = 15$ min) are shown below (Figure 5.10 to Figure 5.13). The HGGA simulation used $\Delta t_x = 15$ min, $\Delta t_{RWC} = 2$ s, $\phi'_1 = 2.0$ m, $\phi'_2 = 0.2$, and $\phi'_3 = 1.0$ m: these tolerances were selected on the basis of a maximum acceptable junction head error of 1.0 m. A comparison of the simulation results illustrates some interesting similarities and differences between the two models. Although the G-GGA failed to capture the up and downsurges, both models predict the same long-term response. This affirms the validity of non-inertial unsteady models for simulating the long-term hydraulics of the WDS here. Nonetheless, some of the transient hydraulics are indeed significant: for example, the downsurge at Zone 1 Pump Station 1 at 22 hours is considerably large. In this case, the assumption of negligible compressibility effects may even be invalid, but of the two models used here, only the HGGA is informative of this.
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

Table 5.2: EPS operations log for pipe network 2

<table>
<thead>
<tr>
<th>Time</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00</td>
<td>Small pump OFF at Z1 PS2</td>
</tr>
<tr>
<td>2:00</td>
<td>Medium pump ON at Z1 PS2</td>
</tr>
<tr>
<td>3:00</td>
<td>Large pump ON at Z1 PS2, medium pump ON at Z1 BPS</td>
</tr>
<tr>
<td>3:15</td>
<td>Medium pump OFF at Z1 PS1, medium pump ON at Z1 PS3, medium pump ON at Z1 PS3, medium pump ON at Z1 PS4, large pump ON at Z1 PS5, small pump OFF at Z1 BPS</td>
</tr>
<tr>
<td>4:00</td>
<td>Small pump OFF at Z1 PS3, small pump OFF at Z1 PS4, small pump OFF at Z1 PS5</td>
</tr>
<tr>
<td>7:30</td>
<td>Medium pump ON at Z1 PS5</td>
</tr>
<tr>
<td>8:00</td>
<td>Medium pump ON at Z1 PS1, medium pump ON at Z1 PS4</td>
</tr>
<tr>
<td>10:00</td>
<td>Large pump ON at Z1 PS2</td>
</tr>
<tr>
<td>14:00</td>
<td>Large pump OFF at Z1 PS1</td>
</tr>
<tr>
<td>18:00</td>
<td>Large pump ON at Z1 PS1</td>
</tr>
<tr>
<td>20:00</td>
<td>Large pump OFF at Z1 PS2, large pump OFF at Z1 PS5, small pump ON at Z1 PS2, small pump ON at Z1 BPS</td>
</tr>
<tr>
<td>21:00</td>
<td>Small pump ON at Z1 PS2, small pump ON at Z1 PS3, medium pump OFF at Z1 PS4, small pump ON at Z1 BPS</td>
</tr>
<tr>
<td>21:15</td>
<td>Medium pump OFF at Z1 PS3, medium pump OFF at Z1 BPS</td>
</tr>
<tr>
<td>22:00</td>
<td>Medium pump OFF at Z1 PS1</td>
</tr>
<tr>
<td>23:00</td>
<td>Medium pump OFF at Z1 PS2</td>
</tr>
</tbody>
</table>

Apart from the simulation results, the other notable difference between the HGGA and conventional EPS is their computational run time. The run time of the HGGA is approximately 12 times greater than that of the G-GGA: this is a consequence of the trade-off between computational efficiency and physical accuracy (i.e., the HGGA’s greater accuracy comes at the expense of a greater run time). Additional factors affecting run time are discussed below.
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

Figure 5.10: Pipe network 2 EPS results – Zone 1 pump station 1 discharge head

Figure 5.11: Pipe network 2 EPS results – Zone 1 pump station 2 discharge head
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

Figure 5.12: Pipe network 2 EPS results – Zone 1 pump station 4 discharge head

Figure 5.13: Pipe network 2 EPS results – Zone 1 elevated tower head
5.6.3 Computational Efficiency

A number of factors affect the computational performance of the HGGA with the adaptive scheme. Key are the simulation duration, $\bar{T}$ (s); the number of operations, $n_{op}$; the average operation period duration, $\bar{T}_{op}$ (s); $\Delta t_x$ and $\Delta t_0$; and the degree of unsteadiness (and thus $\phi_1'$, $\phi_2'$, and $\phi_3'$). One means of characterizing the efficiency of an EPS performed using the HGGA is through an equivalent time step calculated as

$$\Delta t_e = \frac{\bar{T}}{n_{op}(\bar{T}_{op}/\Delta t_0+1)+\bar{T}/\Delta t_x} \tag{5.15}$$

The equivalent time step $\Delta t_e$ (s) is an efficiency indicator. It represents the quotient of the simulation duration and the minimum number of time steps, which would be realized if large indicator tolerances are used. Greater values for $\Delta t_e$ indicate greater efficiency, for this implies fewer computational steps. Equation [5.15] highlights how the analysis parameters affect the solution efficiency. For example, if either $\bar{T}_{op}$ or $n_{op}$ is large, increasing $\Delta t_0$ is the most effective means of improving computational efficiency; however, a more balanced approach might be to increase $\Delta t_x$ and preserve the simulation resolution during transient periods.

To demonstrate $\Delta t_e$, additional EPSs were performed for pipe network 2 using a range of time steps and two sets of indicator tolerances (Table 5.3). The first set of tolerances were assigned large values to ensure no additional microsimulations, and the second set comprises those more practical values used previously. Recall that there are $n_{op} = 15$ operational periods with an average duration of $\bar{T}_{op} = 48$ s.

Table 5.3: HGGA EPS relative run times for pipe network 2

<table>
<thead>
<tr>
<th>Time Steps</th>
<th>Run Times (s)</th>
<th>Relative Run Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t_0$ (s)</td>
<td>$\Delta t_x$ (min)</td>
<td>$\Delta t_e$ (s)</td>
</tr>
<tr>
<td>1.0</td>
<td>7.5</td>
<td>93.2</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>110</td>
</tr>
<tr>
<td>2.0</td>
<td>7.5</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>183</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>204</td>
</tr>
<tr>
<td>3.0</td>
<td>7.5</td>
<td>193</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>246</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>285</td>
</tr>
</tbody>
</table>

$^1\phi_1' = 1 \times 10^4$ m, $\phi_2' = 1 \times 10^4$, and $\phi_3' = 1 \times 10^4$ m

$^2\phi_1' = 2.0$ m, $\phi_2' = 0.2$, and $\phi_3' = 1.0$ m
The results in Table 5.3 illustrate the computational performance of the current work. The equivalent time step correlates well to the HGGA run times, so $\Delta t_e$ is a useful indicator when selecting $\Delta t_0$ and $\Delta t_x$. For all time steps tested, the run time of the HGGA is much greater than that of the G-GGA. This is due to the former’s microsimulations, each of which requires a modest to large amount of iterations to resolve the highly-variable hydraulics during unsteady periods. Comparatively, the G-GGA requires a modest amount of iterations only when operations are performed. Table 5.3 also shows that run time is sensitive to the indicator tolerances. The second set of HGGA simulations has a greater run time, for the smaller indicator tolerances force additional microsimulations as needed, thereby prolonging run time. Nevertheless, this expense is occasionally necessary to avoid overlooking critical transient hydraulics.

5.7 Discussion and Critical Appraisal

Although the HGGA performed well, some challenges remain. Unresolved topics include the appropriate selection of suitable $\phi_1', \phi_2'$, and $\phi_3'$; the practicality of the current methodology vs. that of conventional modeling; and the boundary between compressible and incompressible flow. Alone, the inertial indicators can be used to characterize transient flow. When combined with the HGGA, they inform the solver how to adapt the model to the system, and $\phi_1', \phi_2'$, and $\phi_3'$ are key to this process. The tolerances dictate which dynamic effects are simulated and which are not; therefore, it is important to select appropriate values that reflect the purpose of the analysis. For example, if only the immediate effects of boundary changes are of interest, large tolerances can be adopted (e.g., $\phi_1' = 5$ m, $\phi_2' = 0.5$, and $\phi_3' = 3$ m). Conversely, lower values (e.g., $\phi_1' = 1$ m, $\phi_2' = 0.1$, and $\phi_3' = 0.5$ m) can be used when a more sensitive solution is desired. Results from both examples also showed that indicator tolerances affect computational efficiency. Accordingly, the reader should be mindful that lower tolerances will yield greater physical accuracy, yet also a less efficient solution, making the accuracy-efficiency tension a key concern.

Computational efficiency, physical accuracy, and model complexity should be balanced. The current work strives to achieve this by combining the greater physical accuracy of an RWC solver with the efficiency of a non-inertial unsteady formulation. The present methodology provides a better representation of unsteady hydraulics, yet its use is accompanied by greater complexity, input data requirements, and computational effort. Regarding model input, additional data are required to describe operating protocols in more detail (e.g., valve curve characteristics, pump start and stop controls). Nonetheless, fewer data are required than are needed for transient analyses.
with water hammer models, which also have significantly greater run times and memory requirements. The methodology can be incorporated into optimization schemes, but due to its greater run time, this may be less practical for large networks. Moreover, the HGGA framework can be used to analyze the dynamic implications of planning-level operational decisions and identify critical system states.

A key limitation of the current work is that it cannot simulate boundary changes that invoke compressibility effects. Because it is difficult to characterize compressibility effects, the range of validity of incompressible models is not well known; consequently, it is occasionally necessary to perform simulations with water hammer models to ensure that compressibility effects are indeed not significant. This limitation is not unique to the current work, and it remains difficult to adequately define the significance of compressibility effects despite recent efforts (e.g., Karney, 1990; Wood et al., 1990; Cabrera et al., 1995; Axworthy, 1997; Abreu et al., 1999). Future work on this topic is still needed.

Despite the above-noted limitation, the current work spans the middle ground between the often isolated realms of long-term EPSs and short-term transient analyses. If compressibility effects are indeed present, the HGGA, although possibly invalid, will be more informative of this than a non-inertial unsteady model due to its consideration for inertial effects. Adaptive hybrid modeling can also be used in a similar manner to Filion and Karney’s (2002) work to aid the identification of critical transient states. Overall, this work serves as a step towards developing a comprehensive adaptive hybrid transient model, such as that of Axworthy (1997), yet with more robust solvers and testing with larger and more complex systems.

5.8 Conclusions and Broader Implications

Pressurized transient flow comprises a range of possible behaviours. To better simulate the dynamic incompressible hydraulics of pipe networks, this article presents an adaptive hybrid modeling approach with an adaptive scheme. The novel HGGA formulation combines the G-GGA (Giustolisi et al., 2012) and the RWC GGA (Nault and Karney, 2015), and the adaptive scheme permits gains in computational efficient by varying the time step according to the flow conditions. Three inertial indicators are presented. These characterize the degree of unsteadiness in a network to actively inform the HGGA of how a system should be modeled. A simple system first illustrates the inertial indicators’ utility, while the second example demonstrates a more practical application. Although more computationally intensive than conventional modeling, the
current work was found to provide a more complete representation of dynamic hydraulics.

Despite the advantages of hybrid modeling, one key limitation remains. Incompressible formulations such as the HGGA presume negligible compressibility effects, so their application is limited to simulating boundary changes that do not invoke such effects. This problem is further compounded by the fact that the range of validity of incompressible models are not well known; accordingly, further work is needed to better understand the boundary between compressible and incompressible flow, and thus when each type of model is valid.
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

### Notation

The following symbols are used for scalar terms:

- $\phi_1$: Absolute inertial energy indicator (m)
- $\phi_1'$: Absolute inertial energy indicator tolerance (m)
- $\phi_2$: Relative inertial energy indicator
- $\phi_2'$: Relative inertial energy indicator tolerance
- $\phi_3$: Inertial junction head indicator (m)
- $\phi_3'$: Inertial junction head indicator tolerance (m)
- $\Delta t$: Time step (s)
- $\Delta t_0$: Micro time step (s)
- $\Delta t_x$: Extended time step (s)
- $\Delta t_e$: Equivalent time step (s)
- $\theta$: Momentum equation temporal discretization factor
- $\vartheta$: Nodal continuity integration parameter
- $A_0$: Type 0 node-link incidence matrix
- $A_1$: Type 1 node-link incidence matrix
- $A_{1j}$: Junction-link incidence matrix
- $A_{1t}$: Tank-link incidence matrix
- $A_j$: Area of pipe $j$ (m$^2$)
- $b_0^{(m)}$: Solution vector (m)
- $b_1^{(m)}$: Solution vector (m$^3$/s)
- $b_1^{(t)}$: Solution vector (m$^3$/s)
- $C_1$: Diagonal type 1 node area matrix (m$^2$)
- $D^{(m)}$: Jacobian sub-matrix (s/m$^3$)
- $D_j$: Diameter of pipe $j$ (m)
- $F(*)$: Head loss vector operator (m)
- $G_{1(*)}$: Mass balance vector operator (m$^3$/s)
- $g$: Acceleration due to gravity (m/s$^2$)
- $H_0$: Type 0 node head vector (m)
- $H_1$: Type 1 node head vector (m)
Chapter 5: Adaptive Hybrid Modeling of Incompressible Flows

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{1j}$</td>
<td>$\mathbb{R}^{n_j}$ Junction head vector (m)</td>
</tr>
<tr>
<td>$H_{1T}$</td>
<td>$\mathbb{R}^{n_T}$ Tank head vector (m)</td>
</tr>
<tr>
<td>$h$</td>
<td>$\mathbb{R}^{n_L}$ Head difference vector (m)</td>
</tr>
<tr>
<td>$i$</td>
<td>Start node index</td>
</tr>
<tr>
<td>$j$</td>
<td>Link index</td>
</tr>
<tr>
<td>$k$</td>
<td>End node index</td>
</tr>
<tr>
<td>$L_I$</td>
<td>$\mathbb{R}^{n_L \times n_L}$ Diagonal link inertia matrix ($m^2/s^2$)</td>
</tr>
<tr>
<td>$L_j$</td>
<td>Length of pipe $j$ (m)</td>
</tr>
<tr>
<td>$M^{(m)}$</td>
<td>$\mathbb{R}^{n_L \times n_L}$ Jacobian sub-matrix</td>
</tr>
<tr>
<td>$m + 1$</td>
<td>Current iteration index</td>
</tr>
<tr>
<td>$n_0$</td>
<td>Number of type 0 nodes (i.e., reservoirs)</td>
</tr>
<tr>
<td>$n_1$</td>
<td>Number of type 1 nodes (i.e., junctions and tanks)</td>
</tr>
<tr>
<td>$n_J$</td>
<td>Number of junctions</td>
</tr>
<tr>
<td>$n_L$</td>
<td>Number of links</td>
</tr>
<tr>
<td>$n_{BC}$</td>
<td>Number of boundary change periods</td>
</tr>
<tr>
<td>$n_T$</td>
<td>Number of tanks</td>
</tr>
<tr>
<td>$Q$</td>
<td>$\mathbb{R}^{n_L}$ Link flow vector (m$^3$/s)</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$\mathbb{R}^{n_0}$ Type 0 node outflow vector (m$^3$/s)</td>
</tr>
<tr>
<td>$q_{1(\ast)}$</td>
<td>$\mathbb{R}^{n_1 \rightarrow \mathbb{R}^{n_1}}$ Type 1 node outflow vector operator (m$^3$/s)</td>
</tr>
<tr>
<td>$r^{(t)}$</td>
<td>$\mathbb{R}^{n_L}$ Intermediate solution vector</td>
</tr>
<tr>
<td>$S_I$</td>
<td>Binary inertial solver parameter</td>
</tr>
<tr>
<td>$\bar{T}$</td>
<td>Simulation duration</td>
</tr>
<tr>
<td>$\bar{T}_{BC}$</td>
<td>Average boundary change duration (s)</td>
</tr>
<tr>
<td>$t + 1$</td>
<td>Current time step index</td>
</tr>
<tr>
<td>$V^{(m)}$</td>
<td>$\mathbb{R}^{n_1 \times n_1}$ Intermediate solution matrix</td>
</tr>
</tbody>
</table>
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

Conventional water hammer models are based on the method of characteristics (MOC). The MOC algorithm uses a space-time grid to track the transmission and propagation of waves throughout a network, and hydraulic solutions are computed at each grid point. While reliable and readily implemented, the MOC is more computationally intensive than its alternatives, namely the wave characteristics method (WCM) and algebraic water hammer (AWH). These two methods are not without their own shortcomings though. To benefit from the greater numerical resolution of the MOC and the efficiency of the WCM and AWH, Chapter 6 generalizes the characteristic-based methods to a single formulation that can be implemented as either or a combination of the MOC, WCM, and AWH.

Chapter 6 is based on an article submitted to ASCE’s Journal of Hydraulic Engineering on 2017 March 31. The article was prepared by the author along with Professor Bryan W. Karney and Dr. Bong-Seog (Paul) Jung, and was accepted for publication on 2017-09-05.

6.1 Abstract

Characteristic solution methods, namely the method of characteristics (MOC) and wave characteristics method (WCM), are widely used for simulating transient pipe network flows. Because the MOC computes solutions at interior nodes, it features higher spatial resolution, whereas the WCM makes simplifications that yield more efficient computations. Practical analyses require numerical methods that are both accurate and computationally efficient. To benefit from the advantages of the two approaches, a generalized characteristic method (GCM) is developed here by combining a flexible friction approximation with a variable reach scheme. Significantly, computational savings are realized by selectively providing greater accuracy and higher resolution solutions only where needed via more interior reaches and higher order solutions; further, the new method reduces to either of the MOC and WCM, thereby showing their intrinsic similarities. Multiple examples compare and contrast the numerical methods. From these, unsteady friction effects and, more importantly, spatial resolution are shown to be directly affected by the interior reach treatment, thus exposing a limitation for solution methods with too few interior reaches. Overall, the key contribution of this work is a methodology featuring a similar degree of accuracy to the MOC but with a computational cost better than that of the WCM.
6.2 Introduction

Water pipe systems are designed and operated to meet specific hydraulic requirements for pressure and flow. One particular challenge concerns these properties under transient flow conditions, which arise due to changes at the boundaries of a system (e.g., pump and valve operations). The extreme pressures resulting from rapid changes can lead to pipe breaks, component failure, and contaminant intrusion (Boyd et al., 2004; Friedman et al., 2004), so there is an ever-present need for hydraulic transient analysis via numerical modeling. Though important, models are computationally demanding.

Fundamental to unsteady flow modeling and thus efficient analysis is the numerical solution of the governing water hammer equations. Various approaches have been developed for pipe networks, including Eulerian time domain (Chaudhry and Hussaini, 1985; Zhao and Ghidaoui, 2004; Léon et al., 2008; Chaudhry, 2014), Lagrangian time domain (Ferrante et al., 2009; Huang et al., 2017), and frequency domain schemes (Wylie and Streeter, 1993; Kim, 2007; Zecchin et al., 2010; Vítkovský et al., 2011). Amongst these, the Eulerian method of characteristics (MOC; Wylie and Streeter, 1978) and the Lagrangian wave characteristics method (WCM; Wood et al., 1966) remain the preferred time-domain alternatives due to their ease of implementation, accurate resolution of shock fronts, and ability to handle complex boundary conditions. By considering interior pipe hydraulics, the MOC yields higher resolution solutions; in contrast, the WCM is more efficient due to its underlying simplifications, a key advantage for large pipe networks. Algebraic water hammer (AWH; Wylie and Streeter, 1993; Nault et al., 2016), a meshless variant of the MOC, represents an even further simplification. Each characteristic approach has a distinct advantage and disadvantage.

Central to modeling are numerical accuracy and computational efficiency. The former is necessary to obtain representative solutions. Computational efficiency more so pertains to practical applications, such as problems involving large networks, optimization (Jung et al., 2011), and model calibration (Ebacher et al., 2011). Water distribution system (WDS) analyses, for instance, concern models having thousands to tens of thousands of elements. Without efficient solution methods, analyses must compromise on accuracy or performing fewer simulations. For this reason, many studies emphasize the more efficient WCM as a superior alternative to the MOC (Wood, 2005; Wood et al., 2005a; Ramalingam et al., 2009), but its efficiency comes at the expense of lower solution resolution. Moreover, there is a trade-off between numerical accuracy and computational efficiency.
To improve model efficiency while balancing the need for numerically accurate solutions, this work generalizes the advantages of the MOC, WCM, and AWH into a flexible unified fixed grid characteristic approach for simulating transient pipe network hydraulics. Central to the generalized characteristic method (GCM) are a flexible friction treatment and a variable reach fixed grid scheme; together, these permit individual pipes to be modeled according to the required solution accuracy. More interior reaches and higher order friction treatments are only provided where needed, maintaining numerical accuracy while yielding efficient computations. Multiple networks of varying complexity demonstrate the current work. In each case, the GCM features modest to substantial computational savings, comparable numerical accuracy, and, moreover, greater flexibility than any individual method on its own. Ultimately these developments benefit practical analyses of transient pipe network hydraulics.

6.3 Unsteady Flow Modeling

In formulating the generalized solution, we begin with the characteristic representation of the governing equations. Approximations to the friction integrals are then introduced, and they are combined with the characteristic expressions to assemble a set of compatibility equations that form the solution basis of the GCM.

6.3.1 Governing Equations and Characteristic Representation

One-dimensional unsteady pressurized flow is mathematically described by a pair of non-linear hyperbolic partial differential equations. The momentum equation

\[
\frac{1}{gA} \frac{\partial Q}{\partial t} + \frac{\partial H}{\partial x} + J_{QF} + J_{UF} = 0
\]

represents the effects of fluid inertia and viscous resistance, and the continuity equation

\[
\frac{\alpha^2}{gA} \frac{\partial Q}{\partial x} + \frac{\partial H}{\partial t} = 0
\]

relates to fluid compressibility and conduit elasticity, where \(Q\) = flow (m\(^3\)/s), \(H\) = head (m), \(f\) = friction factor, \(a\) = wave speed (m/s), \(g\) = acceleration due to gravity (m/s\(^2\)), \(A\) = pipe area (m\(^2\)), \(D\) = pipe diameter (m), \(J_{QF}\) = the quasi-steady friction head loss per unit length, \(J_{UF}\) = the unsteady friction head loss per unit length, \(t\) = time (s), and \(x\) = distance along the pipe (m). Derivations and further details can be found in Wylie and Streeter (1993) and Chaudhry (2014).
Characteristic methods transform Equations [6.1] and [6.2] and then integrate them along the positive and negative characteristic lines (Figure 6.1a). This yields the positive compatibility expression

$$B(Q_P - Q_A) + (H_P - H_A) + \int_{x_A}^{x_P} f_{QS} dx + \int_{x_A}^{x_P} f_{US} dx = 0 \quad [6.3]$$

where $B = a/gA$, and the negative compatibility expression

$$B(Q_P - Q_B) - (H_P - H_B) - \int_{x_B}^{x_P} f_{QS} dx - \int_{x_B}^{x_P} f_{US} dx = 0 \quad [6.4]$$

where the subscripts $A, B,$ and $P$ refer to those points shown in Figure 6.1a. Equations [6.3] and [6.4] are respectively valid along the $C+$ and $C-$ lines: these represent the space-time paths along which pressure waves travel.

Figure 6.1: Characteristic representation of wave space-time paths
By discretizing a pipe into \( N_0 \) interior reaches, \( 2N_0 \) characteristic lines and \( N_0 + 1 \) nodes are assembled into a computational grid (Figure 6.1b). Each reach has a length \( \Delta x = L/N_0 \) (m) for a uniform grid. To obtain a stable solution, the discretization must satisfy
\[
C = a \cdot \frac{\Delta t}{\Delta x} \leq 1
\]
which is known as the Courant-Friederich-Lewy (CFL) criterion, where \( C \) = the Courant number, and \( \Delta t \) = the time step (s). Satisfying Equation [6.5] for each pipe in a network poses a challenge, because individual pipes feature different properties. Using constant \( \Delta t \), Eulerian approaches require pipe length or wave speed adjustments to satisfy Equation [6.5]. Interpolation techniques avoid this, though they introduce numerical dispersion. True Lagrangian approaches track wave propagation with variable \( \Delta t \); however, \( \Delta t \) becomes increasingly smaller as wave propagation frequencies increase for long simulation durations in complex networks, thus impeding computational efficiency (Huang et al., 2017). Here, an Eulerian approach is adopted with pipe length and wave speed adjustments. Sufficiently small \( \Delta t \) are thus needed to minimize any such adjustments. The hydraulics of a discretized pipe are then computed by solving the compatibility expressions using a constant \( \Delta t \), but first the friction integrals are resolved.

### 6.3.2 Quasi-Steady Friction Integral Approximations

Viscous fluid resistance is represented by the friction integrals. They are non-linear, so they can only be solved numerically. In the case of the quasi-steady component, the most common approaches are the first order explicit and semi-implicit solutions (Wylie, 1983), each resulting in linear compatibility equations. While efficient, numerical accuracy and stability are also concerned (see Holloway and Chaudhry, 1985; Liou and Wylie, 2014).

To maintain generality, a two-coefficient approximation is used for the quasi-steady friction integrals. That for the positive compatibility expression is
\[
\int_{x_A}^{x_P} J_{QS} \, dx \approx \theta K_P |Q_P| Q_P + (1 - \theta) K_A (\varepsilon |Q_A| Q_P + (1 - \varepsilon) |Q_A| Q_A)
\]
where
\[
K_A = \left( \frac{L}{N_R} \right) \frac{f_A}{2gDA^2}, \quad K_P = \left( \frac{L}{N_R} \right) \frac{f_P}{2gDA^2}
\]
and the friction integral for the negative compatibility expression is
\[
-\int_{x_B}^{x_P} J_{QS} \, dx \approx \theta K_P |Q_P| Q_P + (1 - \theta) K_B (\varepsilon |Q_B| Q_P + (1 - \varepsilon) |Q_B| Q_B)
\]
where
\[
K_B = \left( \frac{L}{N_R} \right) \frac{f_B}{2gDA^2}
\]
where $K = a$ resistance coefficient $(s^2/m^5)$, $0 \leq \theta \leq 1$ and $0 \leq \varepsilon \leq 1 = $ friction integration parameters, and $N_R = $ the number of modeled reaches. The parameter $\theta$ represents a weighting between first order explicit (0) and implicit (1) schemes, where $\theta = 0.5$ yields a second order approximation. Similarly, $\varepsilon = 0$ corresponds to a first order explicit solution, but $\varepsilon > 0$ instead gives a semi-implicit approximation. The number of modeled reaches $N_R$ relates to the discretized number of base reaches $N_0$. Together with $\theta$ and $\varepsilon$, $N_R$ forms a key part of the GCM, a topic that is discussed later.

Equations [6.6] and [6.7] represent the key approximations for characteristic-based methods. Accordingly, their associated numerical error must be controlled to ensure accurate and stable results, and this can be done through the judicious selection of $N_R$, $\theta$, and $\varepsilon$. A first order explicit solution with $\theta = \varepsilon = 0$ is often sufficient. However, in cases with coarse computational grids (low $N_R$), high energy dissipation, or large flow changes, this may not be suitable. Wylie and Streeter’s (1993) evaluative index

$$\psi = \frac{R}{N_R} \ll 1, \quad R = \frac{fL|\bar{Q}|}{2DAa}$$

[6.8]

provides a helpful means of assessing this, where $\bar{Q} =$ a representative (often initial) flow $(m^3/s)$. Equation [6.8] represents the ratio of the head loss to the potential surge magnitude over a single reach. The attenuation index $R$ characterizes the same but for an entire pipe. A first order solution typically yields stable results for $\psi \leq 0.05 – 0.15$ (Wylie and Streeter, 1993); otherwise, a finer discretization (smaller $\Delta t$ and larger $N_R$), higher order solution (with $\theta$ or $\varepsilon > 0$), or implicit solution (also with $\theta$ or $\varepsilon > 0$) is necessary.

In addition to $N_R$, $\theta$, and $\varepsilon$, the treatment of $f$ also affects the solution accuracy. The friction factor can be calculated solely based on the initial conditions using the steady friction approach, or it can be updated throughout a simulation according to the flow conditions (the quasi-steady friction approach). When the flow conditions are highly dynamic, unsteady friction effects become important in the response of a system.

**6.3.3 Unsteady Energy Dissipation**

Unsteady friction is a predominant feature of highly unsteady flows. It arises due to temporal and spatial velocity profile variations, and these typically lead to energy greater dissipation and phase shifts in the frequency response of a system. Numerous unsteady friction models have been developed, from theoretical formulations (Zielke, 1968) to more empirical representations (e.g.,
Brunone et al., 1991). The single coefficient modified instantaneous acceleration-based (MIAB) model of Vítkovský et al. (2006) is considered here due to its wide use in the literature.

Using the MIAB model, the unsteady friction unit head loss term is (Vítkovský et al., 2006)

\[ J_{UF} = \frac{k}{gA} \begin{pmatrix} \frac{\partial Q}{\partial t} + a \cdot \text{sign}(Q) \left| \frac{\partial Q}{\partial x} \right| \end{pmatrix} \]  

where \( k = \) a shear decay coefficient. The parameter \( k \) can be determined experimentally, or it can be estimated empirically as (Vardy and Brown, 1996)

\[ k = \frac{1}{2} \sqrt{c}, \quad c = \begin{cases} 4.76 \times 10^{-3} & \text{Laminar flow} \\ 7.41 \times Re^{-1.0 \log_{10}(14.3 \times Re^{-0.05})} & \text{Turbulent flow} \end{cases} \]

where \( Re = \) the Reynolds number.

Equation [6.9] can be solved in multiple ways. It can be incorporated into the characteristic transform of the governing equations, or the acceleration terms can be represented using finite difference approximations. Here the latter is used due to its compatibility and stability with staggered computational grids (Figure 6.2; see also Vítkovský et al., 2000).

Using the scheme in Figure 6.2, the unsteady friction integrals for the positive and negative characteristic lines are evaluated as

\[ \int_{x_A}^{x_P} J_{UF} \, dx \approx \frac{1}{2} Bk (Q_P - Q_P^{t-2r} + \text{sign}(Q_A)|Q_B - Q_A|) \]  

\[ -\int_{x_B}^{x_P} J_{UF} \, dx \approx \frac{1}{2} Bk (Q_P - Q_P^{t-2r} + \text{sign}(Q_B)|Q_B - Q_A|) \]

respectively, where \( Q_P^{t-2r} = \) a reach-back flow \((\text{m}^3/\text{s})\) at node \( P \) from time step \( t - 2r \), and \( r = N_0/N_R \) is the reach-back index. Together with the compatibility equations and quasi-steady friction integral approximations, the above forms the basis of the generalized approach.

### 6.4 Generalized Characteristic Method

Compared to the MOC and WCM, the GCM represents a flexible approach to modeling transient pipe network hydraulics. First the compatibility expressions are evaluated for a fixed grid with the method of specified intervals, and then solutions thereto are obtained. By combining these with a variable reach scheme, individual pipes are modeled with one or multiple interior reaches to permit flexible and efficient computations.
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

6.4.1 Compatibility Equations

Using the friction integral approximations and unsteady friction model, the compatibility expressions (Equations [6.3] and [6.4]) are expanded. For the positive characteristic, combining Equations [6.3], [6.6], and [6.11] gives

\[
B_A Q_P + H_P - C_A = 0 \tag{6.13}
\]

\[
B_A = \left(1 + \frac{1}{2} k\right) B + \theta K_P |Q_P| + (1 - \theta) \epsilon K_A |Q_A|
\]

\[
C_A = (B - (1 - \theta)(1 - \epsilon)K_A |Q_A|)Q_A + H_A + U_A
\]

\[
U_A = \frac{1}{2} Bk(Q_P^{t-2r} - \text{sign}(Q_A)|Q_B - Q_A|)
\]

where \(B_A\) is a coefficient (s/m²), and \(C_A\) and \(U_A\) are constants (m). Similarly, combining Equations [6.4], [6.7], and [6.12] for the negative compatibility equation leads to

\[
B_B Q_P - H_P - C_B = 0 \tag{6.14}
\]

\[
B_B = \left(1 + \frac{1}{2} k\right) B + \theta K_P |Q_P| + (1 - \theta) \epsilon K_B |Q_B|
\]

\[
C_B = (B - (1 - \theta)(1 - \epsilon)K_B |Q_B|)Q_B - H_B + U_B
\]

\[
U_B = \frac{1}{2} Bk(Q_P^{t-2r} - \text{sign}(Q_B)|Q_B - Q_A|)
\]

where \(B_B\) is a coefficient (s/m²), and \(C_B\) and \(U_B\) are constants (m).

Equations [6.13] and [6.14] are solved for \(Q_P\) and \(H_P\) at each pipe’s interior and end nodes (i.e., junctions). Thus, two types of solutions are needed: interior analyses solve for the interior
pipe hydraulics, and *junction analyses* consider adjacent pipe ends. Pumps and valves are omitted herein, for the topic of this study is the propagation of pressure waves along pipes. Instead, the reader is referred to standard references (e.g., Chaudhry, 2014).

### 6.4.2 Interior and Junction Analyses

Interior analyses concern the black nodes shown in Figure 6.1b. There are only two unknowns, $Q_p$ and $H_p$, and two compatibility equations, one for each adjacent reach (Figure 6.3a), so solutions can be readily obtained. Because $\theta > 0$ may be used, a non-linear solution technique is needed. Applying the Newton-Raphson method to the sum of Equations [6.13] and [6.14] yields

$$Q_p^{(m+1)} = (f^{(m)})^{-1} \left(2\theta K_p^{(m)} \left| Q_p^{(m)} \right| Q_p^{(m)} + C_A + C_B \right)$$

$$f^{(m)} = 4\theta K_p^{(m)} \left| Q_p^{(m)} \right| + (2 + k)B + (1 - \theta)e(K_A|Q_A| + K_B|Q_B|)$$

where $f^{(m)}$ is the Jacobian (s/m²); and $m + 1$ and $m$ = the current and previous solution iterate indices, respectively. By repeatedly solving Equation [6.15] for $Q_p$ until the relative residual

$$\left| Q_p^{(m+1)} - Q_p^{(m)} \right| / \left| Q_p^{(m+1)} \right| < e_{tol},$$

where $e_{tol}$ is an error tolerance, a solution is found. The head $H_p$ is then obtained from Equation [6.13] or Equation [6.14]. If $\theta = 0$, then the compatibility equations are linear, and only a single iteration is needed.

Junction analyses compute the hydraulics at pipe end nodes (shown as the white filled nodes in Figure 6.1b). For an arbitrary junction $i$ with $N_{A(i)} + N_{B(i)}$ adjacent pipes (Figure 6.3b), the associated $N_{A(i)} + N_{B(i)}$ compatibility expressions are combined with one for nodal continuity to solve for $H_{(i)}$. Unlike interior analyses, a more efficient predictor-corrector scheme is used. The initial (predictor) head estimate for a junction $i$ is

$$H_{(i)} = \left( \sum_{j \in N_{A(i)}} \frac{\frac{1}{B_{A(j)}}}{\sum_{j \in N_{A(i)}} \frac{\frac{1}{B_{A(j)}}}} + \sum_{j \in N_{B(i)}} \frac{\frac{1}{B_{B(j)}}}{\sum_{j \in N_{B(i)}} \frac{\frac{1}{B_{B(j)}}} - q_{(i)}} \right)$$

where $q_{(i)}$ = the outflow or demand (m³/s); $N_{A(i)}$ and $N_{B(i)}$ = collections of the link indices for pipes that end (with positive characteristic lines) and start (with negative characteristic lines) at node $i$, respectively; and the terms $B_{A}, B_{B}, C_{A},$ and $C_{B}$ are computed using $\theta = 0$.

Once the predictor solution is calculated, initial flow estimates for the adjacent pipes are obtained from the appropriate compatibility expressions. Then, the final (corrector) solution is computed from Equation [6.16] using the predictor flow estimates and actual $\theta$. Nodal outflows
may be demand-driven, pressure-dependent, or mixed (see Jung et al., 2009b); in such cases, initial and updated estimates of $q(i)$ can be used with the predictor and corrector steps, respectively. As with interior analyses, if $\theta = 0$ for all pipes adjacent to node $i$ and $q(i)$ is demand-driven, it follows that only the predictor junction analysis step needs to be performed because the compatibility equations are linear in $Q_P$. For reference, implementation algorithms for Equations [6.15] and [6.16] are presented in Appendix B.

![Diagram of characteristic lines](image)

**Figure 6.3: Characteristic lines for hydraulic solutions at interior and end nodes**

Before Equations [6.15] and [6.16] can be applied, the parameters $N_R$, $\theta$, and $\varepsilon$ are needed. The first order fixed grid MOC uses $N_R = N_0$ and $\theta = \varepsilon = 0$, meaning hydraulic solutions are computed at each interior node. Large $N_R$ is not always necessary though if a similar solution can be obtained more efficiently by using fewer interior reaches. Often this involves increasing $\Delta t$; though, for pipe networks, increasing $\Delta t$ may result in too coarse of a discretization for some pipes. Instead, $\theta$ and $\varepsilon$ can be selected for individual pipes with $N_R \neq N_0$. 
6.4.3 Variable Reach Scheme

To balance the competing demands of numerical accuracy and computational efficiency, a variable reach scheme (where $N_R \neq N_0$ is permissible) is presented to select $N_R$, $\theta$, and $\epsilon$ for individual pipes. Essentially, a pipe discretized with $N_0$ interior reaches, $C = 1$, and a uniform computational grid can be modeled using any $N_R \in [2, Z(N_0), 2N_0]$, where $Z(N_0)$ denotes the factors of $N_0$. For example, Figure 6.4 shows the possible grids for a pipe with $N_0 = 4$; the staggered grids differ, but the same $\Delta t$ is used to maintain compatibility with adjacent boundary conditions. Values of $\theta = \epsilon = 0$ may suffice for the finer grids (Figure 6.4a and b). Comparatively, the coarser grids (Figure 6.4c and d) are characterized by lower accuracy, meaning $\theta$ or $\epsilon > 0$ are likely more appropriate.

![Figure 6.4: Staggered variable reach computational grids for a pipe with $N_0 = 4$](image)

Two error indices are presented to guide the selection of $N_R$, $\theta$, and $\epsilon$. Consider instantaneous valve closure at time $t = 0$ for the system shown in Figure 6.5, which generates a pressure wave that propagates away from the valve. The head rise at the valve due to the flow change $\Delta Q$ is given by the Joukowsky equation as $\Delta H = -B \cdot \Delta Q$, and from Equation [6.13], the head solution upon valve closure immediately upstream of the valve is $H(t = 0) = C_A$. Neglecting unsteady friction effects, which are typically less important than quasi-steady friction effects in the absence of high-frequency hydraulic variations, the relative error on $\Delta H$ is approximately

$$\epsilon_1 = (1 - W)\psi$$  \[6.17\]

where $W = (1 - \theta)(1 - \epsilon)$ is a weighting factor.
Figure 6.5: Example 1 – idealized reservoir-pipeline-valve system

Following the valve’s closure, \( H(t) \) will increase until it reaches a maximum at time \( T = \frac{2L}{a} \) when the reflected wave front returns. The positive compatibility equation can be used to approximate \( H(t) \) during this period as

\[
H(t) \approx H_1 + B \left( 1 + R \left( \frac{t}{T} - 1 \right) \right) Q_0, \quad 0 \leq t < T
\]

A solution computed using \( N_R, \theta, \) and \( \varepsilon \) will have a maximum head error of approximately \(-2W\Delta t \cdot \partial H / \partial t\) (m), so the head extremum error relative to \( \Delta H \) is

\[
\varepsilon_2 = -W\psi
\]

Equations [6.17] and [6.19] characterize the solution error following sudden flow changes. A value of \( \varepsilon_1 > 0 \) indicates overestimating \( |\Delta H| \), and \( \varepsilon_2 < 0 \) corresponds to underestimating the magnitude of the head extrema. While obtained for downstream flow changes along a single pipe, the expressions are equally representative of upstream flow changes if considered as absolute values; moreover, controlling the error associated with a single pipe also controls that for networks having multiple interconnected pipes.

If \( \varepsilon_1 \) and \( \varepsilon_2 \) are prescribed based on acceptable levels of error, the minimum required \( N_R \) is

\[
N_R \geq R \cdot \max \left\{ \left| \frac{1-W}{\varepsilon_1} \right|, \left| \frac{W}{\varepsilon_2} \right| \right\}
\]

Equation [6.20] directly relates the minimum number of interior reaches to the attenuation index \( R \). This suggests that pipes having greater \( R \) should be modeled with greater \( N_R \), which agrees with the work of others (Wylie and Streeter, 1993; Ramalingam et al., 2009). An efficient solution is obtained by minimizing the right hand side of Equation [6.20], so \( \theta \) and \( \varepsilon \) can be taken as any combination that satisfies

\[
W = \frac{1}{\varepsilon + 1}
\]

where \( E = |\varepsilon_1 / \varepsilon_2| \) is the error tolerance ratio.
Together, the solutions (Equations [6.15] and [6.16]) and variable reach scheme (Equations [6.20] and [6.21]) form the GCM. The error tolerances are simple and intuitive, and they can be chosen for individual pipes according to where greater accuracy is required. Consequently, larger \( N_R \) and more demanding implicit solutions are only provided where needed, thereby facilitating efficient computations. Smaller \( \epsilon_1 \) and \( \epsilon_2 \) provide more accurate solutions, but at the expense of computational efficiency. Altogether, the aim of the GCM is to balance the competing demands of numerical accuracy and computational efficiency.

### 6.5 Computational Accounting

In addition to resolving the accuracy-efficiency tension, the generalized approach encompasses the other numerical methods. Accordingly, it can be implemented as either or a combination of the MOC, WCM, and AWH, which also permits a direct comparison of their computational features. These are discussed more below.

#### 6.5.1 Comparison of Numerical Methods

Between the numerical methods the two key differences are their computational space-time grids (determined by \( N_R \)) and their head loss treatments (represented by \( \theta \) and \( \epsilon \)). Consequently, the GCM reduces to each of the MOC, WCM, and AWH depending on \( N_R \), \( \theta \), and \( \epsilon \). The first order MOC computes hydraulic solutions at all interior nodes, so it is represented by \( N_R = N_0 \) and \( \theta = \epsilon = 0 \). With \( N_R = 2 \), the GCM resembles the WCM by considering the mid-length interaction of impinging pressure waves. Interior head losses are effectively grouped at a single interior node, the so-called friction orifice analogy – additional “friction orifices” are considered with the GCM by increasing \( N_R \). Both first order (\( \theta = 1, \epsilon = 0 \)) and second order (\( \theta = 0.5, \epsilon = 0 \)) implicit approximations have been used with the WCM (Wood et al., 2005b; Ramalingam et al., 2009). Being a derivative of the MOC, AWH considers \( \theta = \epsilon = 0 \) with \( N_R = 1 \) because interior analyses are omitted. Indeed the methods’ computational grids differ, yet each relies on characteristic lines that extend \( N_0/N_R \) time steps back from the solution step (Figure 6.6).
Figure 6.6: Comparison of fixed space-time computational grids without interpolation

While comparing the methods, it is worth further commenting on the GCM and the WCM. Although the generalized approach with $N_R = 2$ is fundamentally the same as the WCM, Equations [6.15] and [6.16] do not reduce to the solution expressions reported in the WCM literature for three reasons. First, the WCM uses opposite sign conventions for flows up and downstream of interior nodes, and Equations [6.15] and [6.16] additionally consider unsteady friction effects. More importantly, the methods’ friction treatments differ: the GCM considers head losses at the intersection of the positive and negative characteristic lines, whereas the WCM concentrates head losses at interior orifice elements (Figure 6.7). Nonetheless, these are merely numerical artefacts.

Indeed, the MOC, WCM, and AWH are profoundly similar. The only differences are their computational grids and friction treatments; otherwise, the methods represent wave propagation in the same manner and thus have similar solution approaches. This contrasts with claims that the MOC and WCM are fundamentally different (Wood et al., 2005a; Ramalingam et al., 2009). Differences between the numerical methods do, however, have implications for numerical accuracy and computational efficiency. Because the GCM can assume the form of each of the MOC, WCM, and AWH, it is used to study the methods’ properties below.
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

6.5.2 Accuracy and Efficiency

Computational efficiency relates to the number of floating point operations (FLOPs) needed to obtain a solution. In turn, operation counts for the GCM are affected by the computational grid (via $N_R$), $\theta$, and $\varepsilon$. To assess these, Table 6.1 lists the interior analysis unit FLOPs for a single pipe (the number of FLOPs per interior node and time step), as well as the minimum required $N_R$ (Equation [6.20]) to provide solutions of comparable accuracy; similarly, the unit FLOPs for junction analyses (per pipe per time step) are reported in Table 6.2. The operation counts are closely based on the interior analysis implementation algorithm presented in Appendix B.
### Table 6.1: Interior analysis unit FLOPs and minimum $N_R$ for different configurations

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$\epsilon$</th>
<th>Minimum $N_R$</th>
<th>Friction Treatment (FLOPs/pipe-node-time step)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Unsteady</td>
</tr>
<tr>
<td>$\theta^*$</td>
<td>$\epsilon$</td>
<td>$R \cdot \max\left(\frac{1-W}{\epsilon_1}, \frac{W}{\epsilon_2}\right)$</td>
<td>103</td>
</tr>
<tr>
<td>$\theta^*$</td>
<td>0</td>
<td>$R \cdot \max\left(\frac{\theta}{\epsilon_1}, \frac{\theta-1}{\epsilon_2}\right)$</td>
<td>98</td>
</tr>
<tr>
<td>0.5$^*$</td>
<td>0</td>
<td>$\frac{1}{2} R \cdot \max\left(\frac{1}{\epsilon_1}, \frac{1}{\epsilon_2}\right)$</td>
<td>89</td>
</tr>
<tr>
<td>$1^*$</td>
<td>0</td>
<td>$R/</td>
<td>\epsilon_1</td>
</tr>
<tr>
<td>0</td>
<td>$\epsilon$</td>
<td>$R \cdot \max\left(\frac{\epsilon}{\epsilon_1}, \frac{\epsilon-1}{\epsilon_2}\right)$</td>
<td>54</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>$\frac{1}{2} R \cdot \max\left(\frac{1}{\epsilon_1}, \frac{1}{\epsilon_2}\right)$</td>
<td>52</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$R/</td>
<td>\epsilon_1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>$R/</td>
<td>\epsilon_2</td>
</tr>
</tbody>
</table>

### Table 6.2: Junction analysis unit FLOPs for the generalized characteristic method

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$\epsilon$</th>
<th>Friction Treatment (FLOPs/pipe-time step)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Unsteady</td>
</tr>
<tr>
<td>$\theta$</td>
<td>$\epsilon$</td>
<td>138</td>
</tr>
<tr>
<td>$\theta$</td>
<td>0</td>
<td>134</td>
</tr>
<tr>
<td>0.5</td>
<td>0</td>
<td>134</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>130</td>
</tr>
<tr>
<td>0</td>
<td>$\epsilon$</td>
<td>116</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>114</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>110</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>110</td>
</tr>
</tbody>
</table>
Table 6.1 and Table 6.2 illustrate how the solution unit operations and minimum required $N_R$ vary the configuration. Fixed $\theta$ and $\varepsilon$ permit modest unit FLOP reductions; more so, the friction treatment largely affects computational effort. For given $\theta$ and $\varepsilon$, an analysis using unsteady friction requires approximately twice as many unit FLOPs as one using steady friction. Additionally, computational effort (the product of the interior analysis unit FLOPs and $N_R - 1$) is equally affected by $N_R$, itself a function of $\theta$ and $\varepsilon$. Consider $[\theta, \varepsilon], [\theta, \varepsilon = 0]$, and $[\theta = 0, \varepsilon]$. These configurations have similar minimum $N_R$ for the same $\varepsilon_1$ and $\varepsilon_2$, so variable $[\theta, \varepsilon]$ is the least efficient treatment due to having the highest interior analysis unit FLOPs. Conversely, $[\theta = 0, \varepsilon]$ is the most efficient while providing comparable flexibility. Key here is that particular reductions permit gains in efficiency without sacrificing numerical accuracy.

Also of interest are the first order MOC ($\theta = \varepsilon = 0$) and second order WCM ($\theta = 0.5, \varepsilon = 0$) representations. If configured to match $\varepsilon_2$ for the MOC (with $N_R = N_0$), the WCM requires $N_R = N_0/2$; thus, the WCM features fewer interior analysis operations for unsteady and quasi-steady friction. In the case of steady friction, however, the MOC requires fewer interior analysis FLOPs when $N_0 > 10$. These remarks do not imply that one method is always more accurate and efficient than the other – such inferences are subject to the specific implementations (see Appendix B) as well as computational effort for junction analyses. Certainly, these discussions are insightful, but the numerical methods’ performance is best compared via example.

### 6.6 Unified Simulation of Transient Flow

Three examples demonstrate the current work. The first, a simple pipeline, tests the performance of the generalized approach for different configurations, and the other two examples emphasize more complex networks. In each case, the method of specified time intervals was used with suitable $\Delta t$ that minimize or avoid the need for length and wave speed adjustments. Further, the WCM was implemented with $N_R = 2$ for simplicity. Vapor cavity formation due to excessive negative pressures is also neglected herein, for the objective is to contrast the numerical methods.

#### 6.6.1 Example 1 – Valve Closure

Unsteady flow analyses are often conducted to estimate the extreme pressures and frequency response resulting from boundary changes. To explore how the generalized solution performs in these respects, instantaneous valve closure was simulated for the system in Figure 6.5 using the GCM with different $N_R$, $\theta$, and $\varepsilon$ – note that $\Delta t$ was selected to eliminate length and wave speed
adjustments. A comparison of the relative head extrema error against $\psi$ (Figure 6.8) illustrates how a finer discretization yields smaller error; further, Figure 6.8 shows that, for the same $\psi$ and $R$, implicit schemes ($\theta > 0$ or $\varepsilon > 0$) better predict the maximum head than first order explicit approximations. This is because $\theta = \varepsilon = 0$ overestimates the friction integral, thereby underestimating the head extrema. Moreover, the results compare well against the $\varepsilon_2$ contours at low $R$. This suggests that Equation [6.18] is rather a reasonable approximation. As $R$ increases though, the error diverges from that predicted by Equation [6.19] due to greater pressure wave attenuation, so the contours provide a conservative overestimate. In addition to the maximum head error, solution resolution is also key.

Adequate solution resolution is necessary to represent frictional pressure wave attenuation. Consider the head predicted upstream of the valve following sudden valve closure (Figure 6.9). The first order explicit MOC uses $N_R = 25$, the GCM is based on the variable reach scheme with $\varepsilon_1 = \varepsilon_2 = 1\%$ (yielding $N_R = 13$), and the WCM considers a second order implementation with $N_R = 2$. Because the WCM uses lower $N_R$, it produces the lowest resolution solution with a maximum head error of -6%. Setting $\theta = 1$ would resolve the latter issue, but the solution resolution can only be improved by increasing $N_R$ at the expense of greater computational effort. The results for the MOC and generalized approach, however, are in much better agreement with the numerically converged solutions due to using sufficiently large $N_R$; in fact, their maximum head errors (-0.9% for the MOC and -1.0% for the generalized method) compare well with $|\varepsilon_2| = 1\%$, as predicted by Figure 6.8a. Key is that the solution error and resolution can be controlled by selecting suitable $\varepsilon_1$ and $\varepsilon_2$ for the GCM.
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

(a) Results for $R = 0.25$

(b) Results for $R = 1$

Figure 6.8: Relative head extrema error for example 1 using the generalized solution
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

Figure 6.9: Comparison of simulation results for example 1 with $R = 0.25$

(a) Results using unsteady friction

(b) Results using quasi-steady friction
6.6.2 Example 2 – Looped Pipe Network

In addition to frictional attenuation, the transient hydraulics of pipe networks are largely affected by the interaction and reflection of waves. With this in mind, example 2 compares the numerical methods as used to simulate sudden valve closure for a small looped pipe network (Figure 6.10; Streeter and Wylie, 1967; the system properties can be readily found in Wood et al., 2005). Pipe length adjustments are minimized to less than 2% by using $\Delta t = 0.02$ s, and the GCM is based on $\epsilon_1 = \epsilon_2 = 1\%$.

![Figure 6.10: Example 2 pipe network (adapted from Streeter and Wylie, 1967)](image)

Though the numerical methods differ, their results for the valve’s upstream head are similar (Figure 6.11). This is a result of the system response being predominantly governed by the interaction and reflection of waves rather than their attenuation; accordingly, the MOC, WCM, AWH, and generalized method yield comparable results due to having similar junction analyses. Indeed, Figure 6.11 does show minor variations between the solutions when unsteady friction effects are considered, a result of differing $N_R$. Not only does $N_R$ affect the solution resolution, it also affects how the finite difference approximations of $\partial Q/\partial t$ and $\partial Q/\partial x$. This is key to adequately capture unsteady energy dissipation, but here the differences are of little importance for practical purposes.
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

(a) Results using unsteady friction

(b) Results using quasi-steady friction

Figure 6.11: Example 2 results for the valve head following sudden valve closure
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

Regarding computational performance, Table 6.3 summarizes the methods’ solution FLOPs per time step. Individual pipes were modeled with \( N_R = 1 \) using the GCM, meaning it has zero interior analysis FLOPs for both friction treatments. Of the three numerical methods, the MOC features the greatest computational effort; in fact, it requires more than 10 times more FLOPs than the GCM, a significant amount. Despite these differences, the three methods provide similar results, with the generalized solution (having the fewest FLOPs) being the most practical.

Table 6.3: Example 2 interior analysis FLOPs per time step

<table>
<thead>
<tr>
<th>Friction Treatment</th>
<th>Solution Method</th>
<th>( N_R )</th>
<th>Computations (FLOPs/time step)</th>
<th>Relative Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Interior Analysis</td>
<td>Junction Analysis</td>
</tr>
<tr>
<td>Unsteady</td>
<td>GCM (( \theta = 0 ))</td>
<td>1.0</td>
<td>0</td>
<td>1,040</td>
</tr>
<tr>
<td></td>
<td>WCM (Second Order)</td>
<td>2.0</td>
<td>800</td>
<td>1,210</td>
</tr>
<tr>
<td></td>
<td>MOC (First Order)</td>
<td>29.2</td>
<td>11,900</td>
<td>990</td>
</tr>
<tr>
<td>Quasi-Steady</td>
<td>GCM (( \theta = 0 ))</td>
<td>1.0</td>
<td>0</td>
<td>720</td>
</tr>
<tr>
<td></td>
<td>WCM (Second Order)</td>
<td>2.0</td>
<td>650</td>
<td>880</td>
</tr>
<tr>
<td></td>
<td>MOC (First Order)</td>
<td>29.2</td>
<td>6,900</td>
<td>670</td>
</tr>
<tr>
<td>Steady</td>
<td>GCM (( \theta = 0 ))</td>
<td>1.0</td>
<td>0</td>
<td>410</td>
</tr>
<tr>
<td></td>
<td>WCM (Second Order)</td>
<td>2.0</td>
<td>330</td>
<td>580</td>
</tr>
<tr>
<td></td>
<td>MOC (First Order)</td>
<td>29.2</td>
<td>3,300</td>
<td>360</td>
</tr>
</tbody>
</table>

6.6.3 Example 3 – Water Distribution System

Example 3 compares the numerical approaches for a more practical application, simulating a power failure event in a WDS (Figure 6.12). An average demand of 25 ML/d is supplied by the system through four pump stations, the primary of which is a high lift facility at a water treatment plant. The model comprises 1,490 pipes and 1,310 junctions. For this example, pipes with diameters greater than 400 mm (typically comprising rigid materials, such as concrete and steel) were assigned \( a = 1,050 \) m/s, and the remaining smaller pipes, often constructed using flexible materials, were assigned \( a = 350 \) m/s. Unlike rigid pipes, flexible pipes typically exhibit viscoelastic deformation effects under highly dynamic conditions. However, this is omitted for simplicity. Power failures are often the most problematic event for WDSs; following the loss of power to a pump, rapid flow deceleration gives rise to highly dynamic conditions. Here the event of interest is a local power failure event at the high lift pump station.
Simulation results from the numerical methods are shown in Figure 6.13 and Figure 6.14. A time step of $\Delta t = 0.02$ s was used due to the otherwise rather high computational cost of using the MOC for a large model. With $\epsilon_1 = \epsilon_2 = 1\%$ and $\theta = 0$, over 90% of the WDS is modeled with $N_R = 1$ for the GCM, whereas the MOC results are based on an average $N_R = 15$. Despite such differences, Figure 6.13 and Figure 6.14 show that the numerical methods yield comparable solutions with only minor variations, again because the system’s response is dominated by pressure wave interaction and reflection. The methods’ solution FLOPs and computational run times, however, differ significantly (Table 6.4). Those for the GCM are less than the solution FLOPs for even the WCM for both friction treatments, making it the preferred alternative.
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

Figure 6.13: Example 3 simulation results with unsteady friction and $\Delta t = 0.02$ s

(a) Head at the high lift pump station discharge

(b) Head at the intermediate high point
Figure 6.14: Example 3 simulation results with steady friction and $\Delta t = 0.02 \text{ s}$
Table 6.4: Example 3 interior analysis FLOPs per time step

<table>
<thead>
<tr>
<th>Friction Treatment</th>
<th>Solution Method</th>
<th>( N_R )</th>
<th>Computations (FLOPs/time step)</th>
<th>Relative Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Interior Analysis</td>
<td>Junction Analysis</td>
</tr>
<tr>
<td>Unsteady</td>
<td>GCM (( \theta = 0 ))</td>
<td>1.4</td>
<td>28,800</td>
<td>173,000</td>
</tr>
<tr>
<td></td>
<td>WCM (Second Order)</td>
<td>2.0</td>
<td>133,000</td>
<td>200,000</td>
</tr>
<tr>
<td></td>
<td>MOC (First Order)</td>
<td>14.7</td>
<td>956,000</td>
<td>164,000</td>
</tr>
<tr>
<td>Steady</td>
<td>GCM (( \theta = 0 ))</td>
<td>1.4</td>
<td>10,700</td>
<td>68,500</td>
</tr>
<tr>
<td></td>
<td>WCM (Second Order)</td>
<td>2.0</td>
<td>55,100</td>
<td>95,300</td>
</tr>
<tr>
<td></td>
<td>MOC (First Order)</td>
<td>14.7</td>
<td>264,000</td>
<td>59,600</td>
</tr>
</tbody>
</table>

### 6.7 Practical Considerations

Examples 1 through 3 demonstrate the superior performance of the generalized approach. Although effective, the variable reach scheme has two shortcomings: Equation [20] does not consider solution stability nor unsteady friction, and using \( \varepsilon = 0 \) may inadvertently zero the friction integral (Liou and Wylie, 2014). These can be addressed by restricting the maximum \( \varepsilon_1 \), \( \varepsilon_2 \), and \( E \), thereby forcing an implicit friction treatment (with greater stability properties; see Chaudhry and Hussaini, 1985) and larger \( N_R \). Simultaneously, larger \( N_R \) also benefit numerical accuracy for unsteady friction. An indicator specific to this would help, but a detailed assessment is beyond the scope of this manuscript. If a pipe is still, however, modeled with \( N_R = 1 \), zeroing the friction integrals is not a major concern, for only pipes with low \( R \) and thus low attenuation potential are represented by \( N_R = 1 \). A minimum \( N_R \) can even be adopted (much like the WCM where \( N_R \geq 2 \)). Even without the aforementioned numerical artefacts, the test cases show that the solutions from the MOC and generalized approach are quite similar.

Considering the approaches’ underlying differences, it is also worth briefly discussing why the generalized method compares well. The answer relates to the adaptive variable reach scheme and, more broadly, the fact that most networks have pipes of low \( R \). For example 1, the system response is predominantly governed by frictional attenuation and thus \( R \). Because the generalized method used sufficiently large \( N_R \) (according to the variable reach scheme), it adequately represents attenuation, whereas the WCM (with \( N_R = 2 \)) does not. Unlike example 1, examples 2 and 3 concern the more common case of networks with pipes having low \( R \). Attenuation thus primarily arises from pressure wave dispersion and demand relief rather than friction; consequently, lower \( N_R \) can be used. It is for this reason that the generalized method and WCM
compare well against the MOC, even though they used much lower $N_R$.

Like the generalized approach, the WCM also uses varying $N_R$ via friction orifices. Though similar, the current work features a number of distinct advantages: $N_R$ is chosen according to $\epsilon_1$ and $\epsilon_2$ rather than heuristics (as in Ramalingam et al., 2009), a variable quasi-steady friction treatment is used, and $N_R = 1$ is permissible. Additionally, unsteady friction has been neglected in the WCM literature but not here, and the GCM solution equations may be reduced for $\theta = 0$ and variable $\epsilon$ to achieve further computational savings with negligible change in accuracy. The main advantage of the GCM is its generality, namely that it can be implemented as either or a combination of the MOC, the WCM, and AWH.

6.8 Conclusions

Practical unsteady flow modeling requires solution methods that are both numerically accurate and computationally efficient. To balance these competing demands, a novel generalized methodology was developed by combining a flexible friction approximation with the variable reach scheme. These allow individual pipes to be modeled according to the required solution accuracy; in this way, larger $N_R$ and higher order friction approximations are only used where needed, thereby facilitating accurate and efficient simulations. Owing to its generality, the present approach encompasses each of the MOC, the WCM, and AWH. From this the characteristic methods are fundamentally similar, with the only differences being $N_R$, $\theta$, $\epsilon$, and, consequently, their computational efficiency. Altogether, the generalized method provides a means of balancing the accuracy-efficiency tension.

Three examples elucidate the current approach. The first explores how $N_R$ affects the solution accuracy and solution resolution for a pipeline system, while the remaining examples concern more complex networks where both numerical accuracy and computational efficiency are key. Even considering unsteady friction, the numerical methods compare well for the pipe networks. And yet, their computational performance, evaluated using interior analysis operation counts, differs significantly. In each case, the generalized method required fewer computations while providing comparable results, a distinct advantage over its alternatives. Ultimately, models are an approximation of reality, so simplifications that permit efficient analyses while adequately representing the underlying physics, such as that here, certainly merit consideration.
Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

Notation

\[
\begin{align*}
\Delta H & \quad \text{Potential surge (m)} \\
\Delta x & \quad \text{Reach length (m)} \\
\Delta t & \quad \text{Time step (s)} \\
\epsilon_1 & \quad \text{Head rise relative error} \\
\epsilon_2 & \quad \text{Extreme head relative error} \\
\epsilon & \quad \text{Friction integration parameter} \\
\theta & \quad \text{Friction integration parameter} \\
\psi & \quad \text{Pipe reach attenuation index} \\
A & \quad \text{Pipe area (m}^2) \\
a & \quad \text{Wave speed (m/s)} \\
B & \quad \text{Pipe compressibility constant (s/m}^2) \\
B_A & \quad \text{Compatibility equation coefficients (s/m}^2) \\
B_B & \quad \text{Compatibility equation coefficients (s/m}^2) \\
C & \quad \text{CFL number} \\
C_A & \quad \text{Compatibility equation constants (m)} \\
C_B & \quad \text{Compatibility equation constants (m)} \\
D & \quad \text{Pipe diameter (m)} \\
E & \quad \text{Error tolerance ratio} \\
f & \quad \text{Friction factor} \\
g & \quad \text{Acceleration due to gravity (m/s}^2) \\
H & \quad \text{Head (m)} \\
J_{QF} & \quad \text{Quasi-steady friction unit head loss} \\
J_{UF} & \quad \text{Unsteady friction unit head loss} \\
K & \quad \text{Resistance coefficient (s}^2\text{/m}^5) \\
k & \quad \text{Shear decay coefficient} \\
L & \quad \text{Pipe length (m)} \\
N_0 & \quad \text{Number of discretized interior reaches} \\
N_{Ai} & \quad \text{Collection of indices for pipes adjacent to junction } i \\
N_{Bi} & \quad \text{Collection of indices for pipes adjacent to junction } i \\
N_R & \quad \text{Number of modeled interior reaches}
\end{align*}
\]
### Chapter 6: Generalized Characteristic Method for Modeling Water Hammer

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q$</td>
<td>Flow (m$^3$/s)</td>
</tr>
<tr>
<td>$q$</td>
<td>Junction outflow (m$^3$/s)</td>
</tr>
<tr>
<td>$R$</td>
<td>Pipe attenuation index</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reynolds number</td>
</tr>
<tr>
<td>$r$</td>
<td>Reach back index</td>
</tr>
<tr>
<td>$t$</td>
<td>Time (s)</td>
</tr>
<tr>
<td>$U_A$</td>
<td>Compatibility equation constants (m)</td>
</tr>
<tr>
<td>$U_B$</td>
<td>Compatibility equation constants (m)</td>
</tr>
<tr>
<td>$W$</td>
<td>Integration weighting factor</td>
</tr>
<tr>
<td>$x$</td>
<td>Longitudinal distance (m)</td>
</tr>
</tbody>
</table>
Chapter 7: Comprehensive Adaptive Hybrid Modeling

Until now, incompressible and unsteady-compressible pipe network hydraulics have been treated separately. This is not, of course, a realistic representation. As discussed in Chapter 2, transient flows vary over a continuous spectrum – dynamic effects are negligible at one end, and at the other they are the dominant flow mechanism. Chapter 7 is the culmination of this thesis. It generalizes the three types of unsteady flow models to a single formulation, the comprehensive global gradient algorithm. The formulation is shown to have a wide physical modeling range; moreover, the modeling framework can be adapted to individual analyses.

As of the writing of this thesis, the content of Chapter 7 has not been published. The author plans to submit the chapter to IAHR’s Journal of Hydraulic Research for publication as a journal article, the title of which being “Comprehensive Modeling of One-Dimensional Unsteady Pipe Network Hydraulics.”

7.1 Abstract

Multiple types of unsteady pipe network flow models exist. Amongst these, water hammer models simulate highly transient hydraulics, whereas incompressible flow formulations are more computationally efficient. To balance the accuracy-efficiency tension, this paper presents a flexible adaptive hybrid transient model (AHTM) capable of simulating the full range of unsteady flow conditions. The methodology combines the comprehensive global gradient algorithm (CGGA), a novel formulation, with unsteady flow characterization indices and an adaptive scheme. Waterhammer, rigid water column, and quasi-steady models are generalized to the CGGA within a unified framework; together with an adaptive flow characterization scheme, the AHTM adjusts the CGGA depending on the degree of unsteadiness. Accordingly, compressibility and inertial effects are only simulated when deemed present using an appropriately small time step, thereby facilitating efficient simulations. The first of two examples illustrates the adaptive ability of the AHTM, and the second concerns the extended period analysis of a water network. From these, the AHTM permits accurate and efficient simulations of unsteady pipe network flow. Moreover, the framework is flexible, allowing it to be tailored to individual analyses.
7.2 Modeling Unsteady Pressurized Hydraulics

Modeling transient pipe network hydraulics presents a complex challenge. In addition to considering arbitrary network topologies and complex unsteady flow phenomena, simulation is complicated by the need for appropriate model selection. Multiple types of models exist, and each has different characteristics (Figure 7.1; see also Abreu et al., 1999 and Jung and Karney, 2016). Physically detailed (and computationally intensive) water hammer type models lie at one end of the spectrum: they simulate highly dynamic conditions where structural integrity is often of concern. At the other end are the more efficient incompressible flow formulations. These include so-called rigid water column (RWC) models, which consider inertial effects, and quasi-steady models, which neglect them. Model selection is only sometimes clear, and it must ultimately consider the purpose of an analysis and the degree of unsteadiness.

![Figure 7.1: Physical accuracy and computational efficiency for pipe network models](image)

Unsteady network hydraulics quite naturally encompass multiple transient flow regimes (Karney, 1990). A model of sufficient physical accuracy is thus necessary for a given analysis, but more physically accurate models have greater computational demand. Accordingly, there is an accuracy-efficiency trade-off. This is practically important, say, for the analysis of water distribution systems (WDSs), design optimization (Jung et al., 2011), and calibration exercises (Ebacher et al., 2011), the latter two of which require multiple simulations. Indeed, a model of suitable physical accuracy is required to obtain representative results. It makes little sense, however, to use that same model throughout an entire simulation when it may only be needed for the most dynamic periods – a reality that motivates adaptive modeling.
Adaptive hybrid transient models (AHTMs) combine multiple types of models and actively use the most suitable one. Axworthy (1997) and later Filion and Karney (2002) are among the first to have introduced AHTMs. Though both were shown to perform well, numerical instabilities are a concern for the RWC component of Axworthy’s work (Axworthy and Karney, 2000; Nault and Karney, 2016a), and Filion and Karney’s model lacks an automatic adaptive component, instead relying on prescribed or user-triggered microsimulations. Moreover, both of the aforementioned perform water hammer simulations with the method of characteristics (MOC; Wylie and Streeter, 1978; Chaudhry, 2014), a less efficient solver than its alternatives (e.g., Wood et al., 2005; Nault et al., 2017). Nault and Karney (2016b) later developed a dual-level AHTM similar to that of Filion and Karney (2002). Their AHTM included an adaptive scheme like Axworthy’s (1997), yet it does not consider unsteady-compressible flow. Each study suffers a limitation, be it numerical stability, computational efficiency, or physical accuracy.

To address the aforementioned challenges, this work presents a general purpose AHTM that combines a novel modeling formulation, unsteady flow characterization indices, and an adaptive scheme. The solver (termed the comprehensive global gradient algorithm or CGGA) encompasses the generalized characteristic method (GCM; Nault et al., 2017) and the hybrid global gradient algorithm (HGGA; Nault and Karney, 2016b), so individual pipes can be modeled using either formulation. Together with unsteady flow characterization indices and an adaptive scheme, the solver and time step are adjusted to suit the flow conditions; in this way, dynamic effects are appropriately invoked, thereby permitting physically accurate yet efficient simulations. For convenience, Table 7.1 contrasts the methodology’s key features against those of existing AHTMs. The current work represents a significant improvement over previous studies regarding flexibility and generality. Two meta-model comparisons are used to illustrate and validate the current AHTM, and conclusions are drawn about the performance of the methodology.

### 7.3 Unsteady Pipe Network Hydraulics

In formulating the CGGA, a pair of hybrid compatibility equations are first obtained from the governing equations. These describe the unsteady compressible-incompressible flow of individual pipes. A network equation is then assembled, and its solution is covered in the following section.
Table 7.1: Comparison of the current AHTM against previous AHTMs

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Quasi-Steady Solver</td>
<td>CGGA as the G-GGA (Giustolisi et al., 2012)</td>
<td>HGGA as the G-GGA (Giustolisi et al., 2012)</td>
<td>GGA (Todini and Pilati, 1988)</td>
<td>Incidence Method (Shimada, 1989) with implicit integration</td>
</tr>
<tr>
<td>RWC Solver</td>
<td>CGGA as the RWC GGA (Nault and Karney, 2016a)</td>
<td>HGGA as the RWC GGA (Nault and Karney, 2016a)</td>
<td>None</td>
<td>Incidence Method (Shimada, 1989) with explicit integration</td>
</tr>
<tr>
<td>Water Hammer Solver</td>
<td>CGGA as the GCM (Nault et al., 2017)</td>
<td>None</td>
<td>MOC (Wylie and Streeter, 1978; Chaudhry, 2014)</td>
<td>MOC (Wylie and Streeter, 1978; Chaudhry, 2014)</td>
</tr>
<tr>
<td>Adaptive Scheme</td>
<td>Event-oriented with physically-based transition indices and variable-length microsimulations</td>
<td>Event-oriented with physically-based transition indices and variable-length microsimulations</td>
<td>Event-oriented with fixed-duration microsimulations</td>
<td>Event-oriented with numerically-based transition indices and variable-length microsimulations</td>
</tr>
</tbody>
</table>

7.3.1 Governing Equations

Mathematically, one-dimensional unsteady flow is described by the momentum equation

$$\frac{1}{gA} \frac{\partial Q}{\partial t} + \frac{\partial H}{\partial x} + J_{QF} + J_{UF} = 0$$

[7.1]

$$J_{QF} = \frac{f Q |Q|}{2gDA^2}$$

$$J_{UF} = \frac{k_{UF} gA}{\partial t} (\frac{\partial Q}{\partial t} + a \cdot \text{sign}(Q) \left| \frac{\partial Q}{\partial x} \right|)$$

and the continuity equation

$$\frac{a^2}{gA} \frac{\partial Q}{\partial x} + \frac{\partial H}{\partial t} = 0$$

[7.2]

where $Q$ = flow (m$^3$/s), $H$ = head (m), $a$ = wave speed (m/s), $g$ = acceleration due to gravity (m/s$^2$), $A$ = pipe area (m$^2$), $D$ = pipe diameter (m), $J_{QF}$ = quasi-steady friction unit head loss, $J_{UF}$ = unsteady friction unit head loss, $f$ = friction factor, $k_{UF}$ = a shear decay coefficient, $t$ = time (s), and $x$ = distance (m) along the pipe (Wylie and Streeter, 1978; Chaudhry, 2014). Equation [7.1] relates to the effects of fluid inertia and viscous resistance, including unsteady friction (Vítkovský et al., 2006). Fluid compressibility and conduit elasticity (so-called “compressibility effects”) are represented by Equation [7.2].
Unlike conventional modeling, no additional physically-based assumptions are made of Equations [7.1] and [7.2]. Instead, a comprehensive formulation is sought. Unsteady-compressible flows are represented by the GCM (Nault et al., 2017), and incompressible flows are handled by the HGGA (Nault and Karney, 2016b). The former, a water hammer solver, generalizes the characteristic-type methods for simulating unsteady hydraulics: for the sake of brevity, GCM interior analyses are not covered here. Similarly, the HGGA generalizes the RWC and quasi-steady modeling formulations of Giustolisi et al. (2012) and Nault et al. (2016a), making it an ideal complement to the GCM. Both the GCM and HGGA are central to the CGGA. Given the formulations’ unique features and varied terminologies, the reader is cautioned that the resulting generalized expressions are rather complex.

7.3.2 Hybrid Compatibility Equations

By applying a characteristic transformation to Equations [1] and [2], the resulting expressions can be combined with a hybrid compressible-incompressible treatment to obtain the negative compatibility equation

\[
S_{C_{jj}} \left[ \frac{1}{g_A} \frac{dQ}{dt} - \frac{1}{a} \frac{dH}{dt} + J_QF + J_{UF} \right] + S_{C_{jj}}' \left[ S_{I_{jj}} \frac{1}{g_A} \frac{\partial Q}{\partial t} + \frac{\partial H}{\partial x} + J_QF \right] = 0
\]  

[7.3]

and the positive compatibility equation

\[
S_{C_{jj}} \left[ \frac{1}{g_A} \frac{dQ}{dt} + \frac{1}{a} \frac{dH}{dt} + J_QF + J_{UF} \right] + S_{C_{jj}}' \left[ S_{I_{jj}} \frac{1}{g_A} \frac{\partial Q}{\partial t} + \frac{\partial H}{\partial x} + J_QF \right] = 0
\]  

[7.4]

for a pipe \( j \), where \( S_{C_{jj}} \) is a binary compressible flow parameter for individual links, \( S_{C_{jj}}' = 1 - S_{C_{jj}} \), and \( S_{I_{jj}} \) is a network-wide binary fluid inertia parameter.

Equations [7.3] and [7.4] are then integrated along a pipe’s characteristic lines. This leads to

\[
B_A \left( Q_{A_{j}}^{(t+1)} \right) - H_{A_{j}}^{(t+1)} + S'_{C_{jj}} H_{B_{j}}^{(t+1)} - S_{C_{jj}} R_{A_{j}} - S'_{C_{jj}} S_{I_{jj}} R_{I_{j}} = 0
\]  

[7.5]

\[
B_A(*) = S_{C_{jj}} \left( B_{j} + \varepsilon_{j} K_{B_{j}} Q_{b_{j}} \right) + \frac{1}{2} B_{j} k_{UF} + S_{C_{jj}}' B_{j} (*)
\]

\[
R_{A_{j}} = \left( B_{j} - (1 - \varepsilon_{j}) K_{B_{j}} Q_{b_{j}} \right) Q_{b_{j}} + H_{B_{j}} + U_{b_{j}}
\]

\[
U_{B_{j}} = \frac{1}{2} B_{j} k_{UF} \left( Q_{A_{j}}^{(t+1-r_{j})} - 2 \cdot \text{sign} \left( Q_{b_{j}} \right) \left| Q_{A_{j}}^{(t+1-r_{j})} - Q_{b_{j}} \right| \right)
\]

and

\[
B_B \left( Q_{B_{j}}^{(t+1)} \right) - S'_{C_{jj}} H_{A_{j}}^{(t+1)} + H_{B_{j}}^{(t+1)} - S_{C_{jj}} R_{B_{j}} - S'_{C_{jj}} S_{I_{jj}} R_{I_{j}} = 0
\]  

[7.6]
Chapter 7: Comprehensive Adaptive Hybrid Modeling

\[ B_B(*) = S_{C_{ij}} \left( B_j + \varepsilon_j K_{A_j} \left| Q_{a_j} \right| + \frac{1}{2} B_j k_{UF} \right) + S'_{C_{ij}} B_i(*) \]

\[ R_{B_j} = (B_j - (1 - \varepsilon_j) K_{A_j} \left| Q_{a_j} \right|) Q_{a_j} + H_{a_j} + U_{a_j} \]

\[ U_{a_j} = \frac{1}{2} B_j k_{UF} \left( Q^{(t+1-\tau_j)}_{B_j} - 2 \cdot \text{sign}(Q_{a_j}) \left| Q^{(t+1-\tau_j)}_{B_j} - Q_{a_j} \right| \right) \]

where

\[ B_i(*) = \frac{1}{\Delta t} S_{I_{jj}} L_{I_{jj}} + F(*) \]

= an incompressible flow head difference operator (m);

\[ R_{I_j} = \frac{1}{\Delta t} L_{I_j} Q^{(t)}_{A_j} - \left( \frac{1-\theta_j}{\theta_j} \right) \left( F \left( Q^{(t)}_{A_j} \right) - H^{(t)}_{A_j} + H^{(t)}_{B_j} \right) \]

= an incompressible flow reach-back term (m) from time step \( t \); \( K_{A_j} = f_{A_j} \Delta x_j / 2 g D_j A_j^2 \) and \( K_B = f_B \Delta x_j / 2 g D_j A_j^2 \) are resistance coefficients (s^2/m); \( Q_{a_j} \) and \( Q_{b_j} \) = interior node reach-back flows (m^3/s) from time step = \( t + 1 - \tau_j \), where \( \tau_j = N_{O_j}/N_{R_j} \), and \( N_{O_j} \) and \( N_{R_j} \) are the number of discretized and modeled interior reaches, respectively; \( L_{I_j} = L_j / g A_j \theta_j \) (s^2/m^2), where \( L_j \) is the pipe length (m); \( F(*) \) = an incompressible flow head loss operator (m); \( \varepsilon_j \) and \( \theta_j \) = friction integration parameters (Karney and McInnis, 1992; Nault and Karney, 2016ab); \( \Delta t \) = time step (s); \( t \) = time step index; and the subscripts \( A \) and \( B \) indicate the start and end of a link, respectively. Unlike Nault et al. (2017), a single coefficient friction integral approximation using \( \varepsilon \) is considered herein for water hammer implementations.

Together, the hybrid compatibility expressions (Equations [7.5] and [7.6]) provide a comprehensive representation of one-dimensional unsteady pipe hydraulics. For unsteady-compressible flow modeled with \( S_{C_{ij}} = 1 \), the expressions describe the start and end conditions of a pipe, and interior analyses are performed using the GCM. If, however, compressibility effects are negligible, then \( S_{C_{ij}} = 0 \) can be used. This reduces both expressions to the incompressible flow momentum equation, which does not require interior analyses (Figure 7.2). Setting \( S_{I_{jj}} = 0 \) further reduces the compatibility equations to the quasi-steady momentum equation. Moreover, \( S_{C_{ij}} \) and \( S_{I_{ij}} \) can actively be selected according to which dynamic effects are important for a given pipe – this is fundamental to adaptive hybrid modeling. In order to solve the compatibility equations though, they are first combined with graph theory notation to form a network equation.
Figure 7.2: Hybrid compressible-incompressible flow representation of an arbitrary pipe $j$

7.3.3 Hybrid Network Equation

Graph theory notation provides a convenient means of describing pipe network topology. An arbitrary pipe network can be described using a node-link incidence matrix $A \in \mathbb{R}^{n_N \times n_L}$ with entries

$$A_{ij} = \begin{cases} +1 & \text{if link } j \text{ starts at node } i \\ 0 & \text{if link } j \text{ is closed or not incident to node } i \\ -1 & \text{if link } j \text{ ends at node } i \end{cases}$$

Nodal boundary conditions are classified according to their features. Type 0 nodes have known head (e.g., reservoirs), and type 1 nodes have unknown head (namely junctions and tanks). From this, the incidence matrix $A$, nodal head vector $H \in \mathbb{R}^{n_N} \text{ (m)}$, and nodal outflow vector $q \in \mathbb{R}^{n_N} \text{ (m}^3/\text{s})$ are partitioned as

$$A^T = \begin{bmatrix} A_0^T & A_1^T \end{bmatrix}$$
$$H^T = \begin{bmatrix} H_0^T & H_1^T \end{bmatrix}$$
$$q^T = \begin{bmatrix} q_0^T & q_1^T \end{bmatrix}$$

where $A_0 \in \mathbb{R}^{n_0 \times n_L}$, $A_1 \in \mathbb{R}^{n_1 \times n_L}$, $H_0 \in \mathbb{R}^{n_0} \text{ (m)}$, $H_1 \in \mathbb{R}^{n_1} \text{ (m)}$, $q_0 \in \mathbb{R}^{n_0} \text{ (m}^3/\text{s})$, $q_1 \in \mathbb{R}^{n_1} \text{ (m}^3/\text{s})$, and the subscripts indicate the respective node type.

Mass conservation for type 1 nodes is given by

$$\frac{\partial}{\partial t} (C_1 H_1) + A_1 Q + q_1 = 0$$

[7.11]
where \( C_1 \in \mathbb{R}^{n_1 \times n_1} \) is a diagonal matrix of type 1 node areas (m\(^2\)), and \( Q \in \mathbb{R}^{n_L} \) is the link flow vector (m\(^3\)/s). Type 1 node outflows may be demand-driven, pressure-dependent, or both, so \( q_1 = q_1(t, H_1) \). Similarly, \( C_1 = C_1(H_1) \) for tanks with level-varying areas. Using a single step variable implicit numerical scheme, Equation [11] can be approximated as (Todini, 2011)

\[
-A_1 Q^{(t+1)} - G_1 \left( H_1^{(t+1)} \right) - b_1 = 0
\]

\[
G_1 \left( H_1^{(t+1)} \right) = q_1 \left( H_1^{(t+1)} \right) + \frac{1}{\vartheta \Delta t} C_1 \left( H_1^{(t+1)} \right) \cdot H_1^{(t+1)}
\]

\[
b_1 = \left( \frac{1-\vartheta}{\vartheta} \right) \left( A_1 Q^{(t)} + q_1 \left( H_1^{(t)} \right) \right) - \frac{1}{\vartheta \Delta t} C_1 \left( H_1^{(t)} \right) \cdot H_1^{(t)}
\]

where \( G_1(*) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_1} \) (m\(^3\)/s), \( b_1 \in \mathbb{R}^{n_1} \) (m\(^3\)/s), and \( \vartheta \) is an integration parameter.

Incompressible flow analyses rely on the compatibility expressions, the network equation is presented above. For unsteady-compressible flow modeling, the following modifications are used (Nault et al., 2016):

\[
\begin{bmatrix}
\bar{A}_0 \\
\bar{A}_1
\end{bmatrix} =
\begin{bmatrix}
A_{0A} & A_{0B} \\
A_{1A} & A_{1B}
\end{bmatrix}
\]

[7.13]

where \( \bar{A}_0 \in \mathbb{R}^{n_0 \times 2n_L} \) and \( \bar{A}_1 \in \mathbb{R}^{n_1 \times 2n_L} \) = compound node-link incidence matrices; \( A_{0A} \) and \( A_{0B} \in \mathbb{R}^{n_0 \times n_L} \) = type 0 node-link incidence matrices comprising the positive and negative elements of \( A_0 \), respectively; and \( A_{1A} \) and \( A_{1B} \in \mathbb{R}^{n_1 \times n_L} \) = type 1 node-link incidence matrices comprising the positive and negative elements of \( A_1 \), respectively. Link start flows are represented by \( Q_A \in \mathbb{R}^{n_L} \) (m\(^3\)/s), and \( Q_B \in \mathbb{R}^{n_L} \) = a link end flow vector (m\(^3\)/s).

By combining the above notation with the compatibility expressions, the network equation

\[
\begin{bmatrix}
B_A(*) & 0 & -A_{1A}^T - S_C A_{1B}^T \\
0 & B_B(*) & -S_C' A_{1A}^T - A_{1B}^T
\end{bmatrix}
\begin{bmatrix}
Q_A^{(t+1)} \\
Q_B^{(t+1)}
\end{bmatrix}
- \begin{bmatrix}
b_A \\
b_B
\end{bmatrix} = 0
\]

[7.14]

\[
b_A = (S_C A_{0A}^T + A_{0B}) H_0 + S_C R_A + S_C' S_I R_I
\]

\[
b_B = (A_{1A}^T + S_C' A_{1B}) H_0 + S_C R_B + S_C' S_I R_I
\]

is assembled, where \( B_A(*) : \mathbb{R}^{n_L} \to \mathbb{R}^{n_L} \) and \( B_B(*) : \mathbb{R}^{n_L} \to \mathbb{R}^{n_L} \) = vector operators (m); \( S_C \in \mathbb{R}^{n_L \times n_L} \) has diagonal elements \( S_{C_{jj}} \); \( S_C' \in \mathbb{R}^{n_L \times n_L} \) has diagonal elements \( S_{C'_{jj}} \); \( S_I \in \mathbb{R}^{n_L \times n_L} \) has diagonal elements \( S_{I_{jj}} \). \( b_A \) and \( b_B \in \mathbb{R}^{n_L} \) (m); \( R_A \) and \( R_B \in \mathbb{R}^{n_L} \) (m); and \( R_I \in \mathbb{R}^{n_L} \) (m).

Similar to the compatibility expressions, the network equation (Equation [7.14]) provides a comprehensive description of unsteady pipe network hydraulics. The first two rows of
Equation [7.12] represent the hybrid compatibility expressions (Equations [7.5] and [7.6]), and type 1 node mass conservation (Equation [7.12]) is enforced by the third row. In addition to interfacing boundary conditions for water hammer, RWC, and quasi-steady models, Equation [7.12] intrinsically considers arbitrarily coupled boundary conditions; consequently, generalized boundary condition treatments (e.g., Axworthy, 1997; Izquierdo and Iglesias, 2004) are not needed. Numerically, Equation [7.14] is a sparse, symmetric, non-linear system of equations. The non-linearity predominantly arises from links modeled with incompressible flow, meaning a non-linear solution method is required. However, because only part of Equation [7.14] is non-linear, there are better strategies than simultaneously solving the entire network equation.

7.4 Comprehensive Global Gradient Algorithm

As a comprehensive solver, the CGGA simulates the full range of transient pipe network hydraulics by solving Equation [7.14]. Because only part of Equation [7.14] is non-linear, element ordering is used to separate the linear and non-linear components. The former is then solved directly, and only the latter is treated iteratively to enable efficient computations.

7.4.1 Element Ordering

Through classifying and sorting a network’s elements, Equation [14] is partitioned into its linear and non-linear components. Type 1 nodes comprise two subsets: independent type 1 (IT1) nodes and dependent type 1 (DT1) nodes. Of these, the latter represent type 1 nodes whose hydraulics are numerically coupled to the hydraulics of another DT1 node via links modeled with incompressible flow. They also include type 1 nodes with non-linear characteristics, such as pressure-dependent outflow. Remaining type 1 nodes are IT1 nodes. A set of linear equations describe the IT1 nodes and their adjacent linearly-related compressible flows (LCFs), whereas the hydraulics of DT1 nodes, their adjacent non-linearly-related compressible flows (NCFs), and incompressible flows are coupled by a non-linear system of equations. To illustrate the node and link classification, the different elements are distinguished in the network below (Figure 7.3; Onizuka, 1986), where only the two valves are modeled with incompressible flow.
Chapter 7: Comprehensive Adaptive Hybrid Modeling

Figure 7.3: Pipe network 1 showing element types (adapted from Onizuka, 1986)

Permutation matrices sort a network’s elements to decouple the linear and non-linear partitions of Equation [7.14]. The permutation matrix $P_1 \in \mathbb{R}^{n_1 \times n_1}$ sorts a network’s type 1 nodes, and $P_A \in \mathbb{R}^{n_L \times n_L}$ and $P_B \in \mathbb{R}^{n_L \times n_L}$ order a network’s link start and end nodes. For type 1 nodes, the permutation matrix $P_1 \in \mathbb{R}^{n_1 \times n_1}$ acts as follows:

$$P_1 H_1 = \begin{bmatrix} H_{IT1} \\ H_{DT1} \end{bmatrix}$$

$$P_1 b_1 = \begin{bmatrix} b_{IT1} \\ b_{DT1} \end{bmatrix}$$

$$P_1 G_1(*)P_1^T = \begin{bmatrix} G_{IT1} & 0 \\ 0 & G_{DT1}(*) \end{bmatrix}$$

where $H_{IT1} \in \mathbb{R}^{n_{IT1}}$ (m); $b_{IT1} \in \mathbb{R}^{n_{IT1}}$ (m$^3$/s); $H_{DT1} \in \mathbb{R}^{n_{DT1}}$ (m); $b_{DT1} \in \mathbb{R}^{n_{DT1}}$ (m$^3$/s); $G_{IT1} \in \mathbb{R}^{n_{IT1} \times n_{IT1}}$ (m$^3$/s); and $G_{DT1}(*) : \mathbb{R}^{n_{DT1}} \to \mathbb{R}^{n_{DT1}} = \text{a vector operator (m$^3$/s)}$.

Similarly, $P_A \in \mathbb{R}^{n_L \times n_L}$ and $P_B \in \mathbb{R}^{n_L \times n_L}$ sort the link start and end flow terms in order of LCFs, NCFs, and incompressible flow as

$$P_A Q_A = \begin{bmatrix} Q_{LCFA} \\ Q_{NCFB} \\ Q_{IF} \end{bmatrix}$$

$$P_B Q_B = \begin{bmatrix} Q_{LCFB} \\ Q_{NCFB} \\ Q_{IF} \end{bmatrix}$$

$$P_A b_A = \begin{bmatrix} b_{LCFA} \\ b_{NCFA} \\ b_{IF} \end{bmatrix}$$

$$P_B b_B = \begin{bmatrix} b_{LCFB} \\ b_{NCFB} \\ b_{IF} \end{bmatrix}$$

where $Q_{LCFA} \in \mathbb{R}^{n_{LCFA}}$ (m$^3$/s); $b_{LCFA} \in \mathbb{R}^{n_{LCFA}}$ (m); $Q_{LCFB} \in \mathbb{R}^{n_{LCFB}}$ (m$^3$/s); $b_{LCFB} \in \mathbb{R}^{n_{LCFB}}$ (m); $Q_{NCFA} \in \mathbb{R}^{n_{NCFA}}$ (m$^3$/s); $b_{NCFA} \in \mathbb{R}^{n_{NCFA}}$ (m); $Q_{NCFB} \in \mathbb{R}^{n_{NCFB}}$ (m$^3$/s);
\( b_{NCF_B} \in \mathbb{R}^{n_{NCF_B}} \) (m); and \( Q_{IF} \in \mathbb{R}^{n_{IF}} \) (m/s); and \( b_{IF} \in \mathbb{R}^{n_{IF}} \) (m). Note that \( Q_{IF} \) and \( b_{IF} \) arise in both expressions since the start and end flow are equivalent for a link modeled with incompressible flow. Similar to \( Q_A \) and \( Q_B \), the operators \( B_A(*) \) and \( B_B(*) \) are sorted as

\[
P_A B_A(*) P_A^T = \begin{bmatrix} B_{LCF_A} & 0 & 0 \\ 0 & B_{NCF_A} & 0 \\ 0 & 0 & B_{IF(*)} \end{bmatrix} \quad [7.19]
\]

and

\[
P_B B_B(*) P_B^T = \begin{bmatrix} B_{LCF_B} & 0 & 0 \\ 0 & B_{NCF_B} & 0 \\ 0 & 0 & B_{IF(*)} \end{bmatrix} \quad [7.20]
\]

where \( B_{LCF_A} \in \mathbb{R}^{n_{LCF_A} \times n_{LCF_A}} \) (m), \( B_{LCF_B} \in \mathbb{R}^{n_{LCF_B} \times n_{LCF_B}} \) (m), \( B_{NCF_A} \in \mathbb{R}^{n_{NCF_A} \times n_{NCF_A}} \) (m), \( B_{NCF_B} \in \mathbb{R}^{n_{NCF_B} \times n_{NCF_B}} \) (m), and \( B_{IF(*)} : \mathbb{R}^{n_{IF}} \rightarrow \mathbb{R}^{n_{IF}} \) is a vector operator (s/m²).

Element ordering is also applied to the incidence matrices. They are ordered as

\[
P_1 A_1 A_1 P_1^T = \begin{bmatrix} A_{LCF_A} & 0 & 0 \\ 0 & A_{NCF_A} & A_{IF_A} \\ 0 & A_{NCF_B} & A_{IF_B} \end{bmatrix} \quad [7.21]
\]

\[
P_1 A_1 B_1 P_1^T = \begin{bmatrix} A_{LCF_A} & 0 & 0 \\ 0 & A_{NCF_A} & A_{IF_A} \\ 0 & A_{NCF_B} & A_{IF_B} \end{bmatrix} \quad [7.22]
\]

where \( A_{LCF_A} \in \mathbb{R}^{n_{IT1} \times n_{LCF_A}} \), \( A_{LCF_B} \in \mathbb{R}^{n_{IT1} \times n_{LCF_B}} \), \( A_{NCF_A} \in \mathbb{R}^{n_{DT1} \times n_{NCF_A}} \), \( A_{NCF_B} \in \mathbb{R}^{n_{DT1} \times n_{NCF_B}} \), and \( A_{IF_A} \) and \( A_{IF_B} \) \( \in \mathbb{R}^{n_{DT1} \times n_{IF}} \). For incompressible flows, \( A_{IF_A} \) and \( A_{IF_B} \) can be simplified as \( A_{IF} = A_{IF_A} + A_{IF_B} \in \mathbb{R}^{n_{DT1} \times n_{IF}} \).

With these, Equation [7.14] is manipulated into two independent systems of equations. The linear part

\[
\begin{bmatrix} \bar{B}_{LCF} & -\bar{A}_{LCF} \\ -\bar{A}_{LCF} & -G_{IT1} \end{bmatrix} \begin{bmatrix} \bar{Q}^{(t+1)}_{LCF} \\ H_{IT1}^{(t+1)} \end{bmatrix} - \begin{bmatrix} \bar{b}_{LCF} \\ \bar{b}_{IT1} \end{bmatrix} = 0 \quad [7.23]
\]

\[
\bar{Q}^{(t+1)}_{LCF} = \begin{bmatrix} Q^{(t+1)}_{LCF_A} \\ Q^{(t+1)}_{LCF_B} \end{bmatrix} \quad \bar{b}_{NBICF} = \begin{bmatrix} b_{LCF_A} \\ b_{LCF_B} \end{bmatrix}
\]

\[
\bar{B}_{LCF} = \begin{bmatrix} B_{LCF_A} & 0 \\ 0 & B_{LCF_B} \end{bmatrix} \quad \bar{A}_{LCF} = \begin{bmatrix} A_{LCF_A} & A_{LCF_B} \end{bmatrix}
\]

represents the hydraulics of IT1 nodes and their adjacent LCFs, where \( \bar{Q}^{(t+1)}_{LCF} \in \mathbb{R}^{n_{LCF}} \) (m³/s);
\[ \bar{b}_{LCF} \in \mathbb{R}^{n_{LCF}} \text{ (m)}; \quad \bar{B}_{LCF} \in \mathbb{R}^{n_{LCF} \times n_{LCF}} \text{ (m)}; \quad \bar{A}_{LCF} \in \mathbb{R}^{n_{IT1} \times n_{LCF}}. \] The non-linear complement to Equation \([7.23]\) describes DT1 nodes, NCFs, and incompressible flows:

\[ \begin{bmatrix} \bar{B}_{NL}(*) & -\bar{A}_{NL} \\ -\bar{A}_{NL} & -G_{DT1}(*) \end{bmatrix} \begin{bmatrix} Q_{NL}^{(t+1)} \\ H_{DT1}^{(t+1)} \end{bmatrix} - \begin{bmatrix} \bar{b}_{NL} \\ b_{DT1} \end{bmatrix} = 0 \quad \text{[7.24]} \]

\[ \bar{B}_{NF}(*) = \begin{bmatrix} B_{NCF_A} & 0 & 0 \\ 0 & B_{NCF_B} & 0 \\ 0 & 0 & B_{IF}(*) \end{bmatrix}, \quad \bar{A}_{BF} = [A_{NCF_A}, A_{NCF_B}, A_{IF}] \]

where \( Q_{NL}^{(t+1)} \in \mathbb{R}^{n_{BF}} \text{ (m}^3/s); \quad \bar{b}_{NL}(*) \in \mathbb{R}^{n_{BF}} \text{ (m);} \quad \bar{B}_{NL}(*) : \mathbb{R}^{n_{NCF} + n_{IF}} \rightarrow \mathbb{R}^{n_{NCF} + n_{IF}} \) is a vector operator (m); and \( \bar{A}_{NL} \in \mathbb{R}^{n_{IT1} \times n_{NCF} + n_{IF}}. \] By sorting a network’s elements, \( S_C \) and \( S'_C \) are dropped, and the network equation can be solved by considering Equations \([7.23]\) and \([7.24]\).

### 7.4.2 Solutions to the Component Expressions

With the network expression separated into its linear and non-linear components, solutions can be obtained via appropriate techniques. Equation \([7.23]\), the linear part, has a saddle point structure (Benzi et al., 2005), so it can be solved via Schur complement reduction as

\[ H_{IT1}^{(t+1)} = -V_{IT1}^{-1}(\bar{A}_{LCF}\bar{B}_{LCF}^{-1}\bar{b}_{LCF} + b_{IT1}) \quad \text{[7.25]} \]

\[ \bar{Q}_{LCF}^{(t+1)} = \bar{B}_{LCF}^{-1}(\bar{A}_{LCF}H_{IT1}^{(t+1)} + \bar{b}_{LCF}) \]

\[ V_{IT1} = \bar{A}_{LCF}\bar{B}_{LCF}^{-1}\bar{A}_{LCF}^T + G_{IT1} \]

where \( V_{IT1} \in \mathbb{R}^{n_{IT1} \times n_{IT1}} \text{ (m}^2/s). \) Because each column within \( \bar{A}_{LCF} \) contains at most one non-zero value, \( V_{IT1} \) is a diagonal matrix. In fact, Equation \([7.25]\) is simply a matrix representation of the GCM’s junction analysis, so \( H_{IT1}^{(t+1)} \) and \( \bar{Q}_{LCF}^{(t+1)} \) can be computed directly without the need for factorization.

Unlike Equation \([7.23]\), the non-linear Equation \([7.24]\) cannot be solved directly. Instead, the Newton-Raphson method is applied to obtain the recursive solution

\[ \begin{bmatrix} M^{(m)} & -\bar{A}_{NL}^T \\ -\bar{A}_{NL} & -g_{DT1}^{(m)} \end{bmatrix} \begin{bmatrix} \bar{C}_{NL}^{(m+1)} \\ H_{DT1}^{(m+1)} \end{bmatrix} = \begin{bmatrix} \bar{b}_{NL}^{(m)} \\ b_{DT1}^{(m)} \end{bmatrix} \quad \text{[7.26]} \]
Chapter 7: Comprehensive Adaptive Hybrid Modeling

\[ \bar{b}_{NL}^{(m)} = M^{(m)} \bar{q}_{NL}^{(m)} - \bar{B}_{NL} \bar{q}_{NL}^{(m)} + \bar{b}_{NL} \]

\[ b_{DT1}^{(m)} = g_{DT1}(H_{DT1}^{(m)} - g_{DT1}^{(m)} H_{DT1}^{(m)} + b_{DT1}^{(m)}) \]

where \( M^{(m)} \in \mathbb{R}^{n_{NCF}+n_{IF} \times n_{NCF}+n_{IF}} \) is the Jacobian of \( \bar{B}_{NL}(\bar{Q}_{NL}^{(m)}) \) with respect to \( \bar{Q}_{NL} \) (s/m²); \( g_{DT1}^{(m)} \in \mathbb{R}^{n_{DT1} \times n_{DT1}} \) is the Jacobian of \( G_{DT1}(H_{DT1}^{(m)}) \) with respect to \( H_{DT1} \) (m²/s); \( \bar{b}_{NL}^{(m)} \in \mathbb{R}^{n_{NL}} \) (m); \( b_{DT1}^{(m)} \in \mathbb{R}^{n_{DT1}} \) (m³/s); and \( m + 1 \) is the current iterate.

Equation [7.26] features a saddle point structure similar to Equation [7.23]. Its solution is thus

\[ H_{DT1}^{(m+1)} = -(V_{DT1}^{(m)})^{-1} (A_{BF}(M^{(m)})^{-1} b_{BF}^{(m)} + b_{DT1}^{(m)}) \]  

\[ \bar{Q}_{NF}^{(m+1)} = (M^{(m)})^{-1} (A_{NF}^{T} H_{DT1}^{(m+1)} + b_{NF}^{(m)}) \]

\[ V_{DT1}^{(m)} = A_{NF}^{T}(M^{(m)})^{-1} A_{NF}^{T} + g_{DT1}^{(m)} \]

where \( V_{DT1}^{(m)} \in \mathbb{R}^{n_{DT1} \times n_{DT1}} \) (m²/s). To compute \( H_{DT1}^{(m+1)} \), an \( \mathbb{R}^{n_{DT1} \times n_{DT1}} \) system of equations must be solved. Given that \( V_{DT1}^{(m)} \) is positive definite, methods such as Cholesky decomposition or the iterative conjugate gradient method can be applied. For the latter, iterations can be terminated prematurely once the residual to Equation [7.27] is sufficiently small, yielding an inexact Newton method (Dembo et al., 1980).

A solution to Equation [7.26] is obtained by successively solving Equation [7.27]. A typical convergence criterion for the Newton-Raphson iterations is

\[ \| r^{(m)} \|_2 < \epsilon \| b \|_2, \quad b = [\bar{B}_{NL}^{T} \quad b_{DT1}^{T}]^{T} \]

where \( r^{(m)} \in \mathbb{R}^{n_{NF}+n_{DT1}} \) is the residual error to Equation [7.24], \( b \in \mathbb{R}^{n_{NF}+n_{DT1}} \) (m | m³/s), and \( \epsilon \) = a convergence tolerance.

To summarize a rather abstract set of relations, a solution to the hybrid network expression (Equation [7.22]) is obtained by solving its linear and non-linear components. When performing a water hammer simulation, interior hydraulics are computed using the GCM’s interior analysis, and both Equations [7.25] and [7.27] are used. Comparatively, only Equation [7.27] is solved for incompressible flow simulations. By generalizing the GCM (Nault et al., 2017) and HGGA (Nault and Karney, 2016b), the CGGA can simulate one-dimensional pipe network hydraulics ranging from quasi-steady to unsteady-compressible; moreover, the CGGA’s configuration and \( \Delta t \) may
be changed throughout a simulation depending on whether dynamic effects are important or inconsequential, a key feature for adaptive modeling. This is the subject of the following section.

### 7.5 Adaptive Unsteady Flow Modeling

Adaptive hybrid modeling strives to balance the competing objectives of physical accuracy and computational efficiency. Essentially, model types with greater physical accuracy (and thus greater computational demand) are only applied when warranted. There are three components to the current AHTM, the first of which (the CGGA) has already been covered. Unsteady flow characterization indices are presented below to guide the selection of \( S_c, S_I, \) and \( \Delta t \); the indices are then combined with an adaptive scheme that relates them to the CGGA.

#### 7.5.1 Unsteady Flow Characterization

Parallel to the different types of models are the transient flow regimes (Karney, 1990). Each is characterized by its predominant dynamic features – for example, compressibility and inertial effects (and sometimes unsteady friction) define unsteady-compressible flow conditions. From early works (e.g., Allievi, 1913) to more recent efforts (Karney, 1990; Abreu et al., 1999; Jung and Karney, 2016), many have studied the flow regime boundaries and thus the conditions in which each type of model should be applied. But rather than distinct boundaries, there are transitional regions that depend on the flow physics and network properties (Figure 7.4). Understanding this is key to appropriate model selection.

![Figure 7.4: Transient flow regimes and relative importance of dynamic effects](image)

Within the context of adaptive modeling, indicators are introduced to actively characterize dynamic effects throughout a simulation and aid model selection. Suitable indicators can be
obtained from the governing equations upon which each model is based. Consider the positive
compatibility expression (Equation [7.4]), but without the binary solver parameters nor unsteady
friction terms. Under nearly incompressible flow conditions, integrating this expression along the
length of a pipe \( j \) yields the approximation

\[
\frac{L_j}{2a_j} \left( \frac{\partial H_{1j}}{\partial t} + \frac{\partial H_{2j}}{\partial t} \right) + \frac{L_j}{g A_j} \frac{dQ_j}{dt} + F(Q_j) - h_j \approx 0
\]

[7.29]

where \( h_j = H_{1j} - H_{2j} \) is the pipe’s head difference, \( H_{1j} \) is the upstream head (m), and \( H_{2j} \) is the
downstream head (m).

Similar to Karney’s (1990) integrated energy expression, Equation [7.29] features two terms
for dynamic effects, one for viscous resistance, and a fourth for link end conditions. The
magnitude of dynamic effects is indicated by the first two terms in Equation [7.29]: physically,
these represent an imbalance to otherwise quasi-steady flow conditions where \( F(Q_j) - h_j = 0 \).
The absolute dynamic effects indicator

\[
\phi_{A_j} = \frac{L_j}{2a_j} \left| \frac{\partial H_{1j}}{\partial t} + \frac{\partial H_{2j}}{\partial t} \right| + \frac{L_j}{g A_j} \left| \frac{dQ_j}{dt} \right|
\]

[7.30]

characterizes this imbalance, where \( \phi_A \in \mathbb{R}^{nL} \) (m). The first term in Equation [7.30] represents
the product of half of the wave travel time (\( 2L/a \)) and the average time rate of change in head at
a pipe’s ends, while the second term, the acceleration head, indicates the magnitude of inertial
effects. Large values for the elements of \( \phi_A \) imply that dynamic effects are not negligible. A
quasi-steady model likely represents a pipe \( j \) sufficiently if \( \phi_{Aj} \) is small; otherwise, the assumption
of negligible dynamic effects is invalid, and a more physically accurate type of model is necessary.

Though Equation [7.30] is insightful and physically-based, it is biased by pipe lengths. There
also remains the question of “how large is large?” To assess the importance of dynamic effects,
the relative magnitude of the first three terms on the left hand side of Equation [7.29] can be
compared to obtain

\[
\phi_{Rj} = \frac{\phi_{A_j}}{\phi_{A_j} + |F(Q_j)|}
\]

[7.31]

where \( \phi_R \in \mathbb{R}^{nL} \) = the dimensionless relative dynamic effects indicator.

Equations [7.30] and [7.31] characterize the unsteadiness of individual pipes. Consistently
large indicators suggest that dynamic effects are present and important. If \( \phi_{A_j} \) and \( \phi_{Rj} \) are both
greater than some tolerances, then an RWC or water hammer model should be used; otherwise, a quasi-steady model should suffice. To this end, the absolute tolerances $\phi'_{AI}$ and $\phi'_{AD}$ and relative tolerances $\phi'_{RI}$ and $\phi'_{RD}$ are introduced, where the subscripts $I$ and $D$ respectively indicate inertial and dynamic (i.e., inertial plus compressibility) effects. The tolerances relate to the transitional regions in Figure 7.4. Indeed, specifying tolerances is analogous to flow regime delineation, but the current approach features two advantages – *a priori* information about a system’s hydraulics is not needed, and the tolerances may be chosen according to which dynamic effects are of interest for a given analysis. The indicators are thus suitable for use with the CGGA.

### 7.5.2 Adaptive Modeling Scheme

With the CGGA and transient flow indicators, the third component of the AHTM is the adaptive scheme. It selects $S_C$, $S_I$, and $\Delta t$ according to $\phi_A$ and $\phi_R$, so dynamic effects are only simulated when deemed present to facilitate efficient simulations. Dynamic effects arise due to boundary condition changes; similar to past work (Nault and Karney, 2016b; Filion and Karney, 2002), the adaptive scheme here considers an event-oriented approach (Figure 7.5). At the beginning of an extended period, the solution is advanced to immediately before a boundary condition change via a *quasi-steady front end simulation*. A series of *midway microsimulations* are then performed using much smaller $\Delta t$: these capture a system’s resulting dynamic response. The microsimulations continue until the later of when the event is completed or when dynamic effects dissipate, which is determined by the indicator tolerances. Finally, a *back end quasi-steady simulation* advances the solution to the beginning of the next extended period. In essence, the adaptive scheme uses model types of greater physical accuracy only when needed.

*Figure 7.5: Event-oriented adaptive implementation of the CGGA*
During the midway period, microsimulations are performed while actively adjusting the CGGA and $\Delta t$ according to $\phi_A$ and $\phi_R$. Because $\Delta t$ is consistent throughout a network, it is sensible to use the same type of model throughout a network. Thus, the elements of $S_C$ and $S_I$ are taken as

$$S_{C_{jj}} = \begin{cases} 1 & \text{If } \phi_{A_k} > \phi_{A_D} \text{ and } \phi_{R_k} > \phi_{R_D} \text{ for any pipe } k, j \in P_C \\ 0 & \text{Otherwise} \end{cases}$$

[7.32]

and

$$S_{I_{jj}} = \begin{cases} 1 & \text{If } \phi_{A_k} > \phi_{A_I} \text{ and } \phi_{R_k} > \phi_{R_I} \text{ for any pipe } k \\ 0 & \text{Otherwise} \end{cases}$$

[7.33]

where $P_C = \text{the collection of pipes modeled with compressible flow when the CGGA is implemented as a water hammer model.}$ Some pipes (the collection of which denoted by $j \in P_I$) may be modeled solely with incompressible flow, such as short pipes whose wave travel time may be significantly distorted if represented by a water hammer model. Unlike pipes, pumps and valves are only modeled using $S_{C_{jj}} = S_{I_{jj}} = 0$. Suitable $\Delta t$ is also necessary for numerical accuracy and numerical stability. For water hammer implementations, a time step of $\Delta t_{WH} = 10^{-3}$ to $10^{-2}$ is typical, and $\Delta t_{WH}$ must satisfy stability criteria (see Chaudhry, 2014). Time steps of $\Delta t_{RWC} = 10^{-1}$ to $10^1$ s and $\theta = 1$ are ideal for unsteady-incompressible flow analyses (Nault and Karney, 2016ab). Lastly, nearly any $\Delta t$ will yield stable results for quasi-steady flow simulations, provided $\theta$ is close to unity (Todini, 2011). These all ensure stable and representative results.

Beyond the above, a few other artefacts are adopted. The first microsimulation during a midway period uses the CGGA as a water hammer model, so instantaneous, fast, and slow events do not need to be distinguished. During midway periods when the CGGA is implemented as an RWC model yet compressibility effects are deemed present, the microsimulation is redone using a water hammer model; moreover, to avoid cycling between model types and ensure smooth transitions, additional water hammer and RWC microsimulations are forced after dynamic (compressibility and inertial) and inertial effects have dissipated, respectively. Lastly, two sets of incidence matrices are pre-allocated for speed. One set is for incompressible flow solves, and the other is for cases where the CGGA is implemented as a water hammer model. Pseudocode for the AHTM’s overall implementation is shown below (Figure 7.6).
1. **Front End Simulation:** At beginning of extended period $i$ at time $T_i$ with an event at time $T_{i,1}$, perform a front end simulation:

   \[ t = 0 \]
   
   MODE = ‘quasi-steady’
   \[ \Delta t = T_{i,1} - T_i - \Delta t_{WH} \] and $T_{i,t} = T_i + \Delta t$

   Calculate the solution at time $T_{i,t}$

2. **Midway Microsimulations:**

   MODE = ‘water hammer’
   \[ \Delta t = \Delta t_{WH} \]

   While true:
   
   \[ t = t + 1 \text{ and } T_{i,t} = T_{i,t-1} + \Delta t \]

   Calculate the solution at time $T_{i,t}$

   If MODE = ‘water hammer’:
   \[ f_{WH} = f_{WH} - 1 \]

   If $\max(\phi_{A_j} < \phi_{A_D}^{'}$ or $\phi_{R_j} < \phi_{R_D}^{'}$ for any pipe $j$ and
   \[ f_{WH} \leq 0: \]
   
   MODE = ‘RWC’
   \[ \Delta t = \Delta t_{RWC} \]

   Else if MODE = ‘RWC’:
   \[ f_{WH} = f_{WH0} \]

   If $\max(\phi_{A_j} > \phi_{A_D}^{'}$ and $\phi_{R_j} > \phi_{R_D}^{'}$ for any pipe $j$:
   
   MODE = ‘water hammer’
   \[ t = t - 1 \text{ and } \Delta t = \Delta t_{WH} \]

   Else if $T_{i,t} \geq T_{i,1} + \Delta T_{op}$ and $\max(\phi_{A_j} < \phi_{A_D}^{'}$ or $\max(\phi_{R_j} < \phi_{R_D}^{'}$ for any pipe $j$, end loop

3. **Back End Simulation:**

   MODE = ‘quasi-steady’
   \[ \Delta t = T_{i+1} - T_{i,t} \]

   Calculate the solution at time $T_{i+1}$

---

Figure 7.6: Comprehensive adaptive hybrid transient model implementation

### 7.6 Comprehensive Simulation of Unsteady Flows

Verification and validation are necessary for computational models to demonstrate their adequacy (Oberkampf and Trucano, 2002; Roy and Oberkampf, 2011). However, only meta-model comparisons are considered here, where a subject model, the comprehensive AHTM, is contrasted against proven models. The present subject matter more so concerns the representation of the underlying flow physics; further, uncertainties vary widely by analyses and the particular network, making meta-model comparisons preferrable. Two examples demonstrate the comprehensive AHTM. The first, a small looped pipe network, validates the current work against an MOC-based water hammer model, while the second illustrates a more practical application,
the 24-hour extended period simulation (EPS) of a water transmission system. Simulations were performed with $\theta = \vartheta = 1$, unsteady friction during water hammer solves, GCM discretization error tolerances of 1\%, and the parameters summarized in Table 7.2.

\textit{Table 7.2: Simulation parameters for time steps and indicator tolerances}

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t_{WH}$ (s)</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$\Delta t_{RWC}$ (s)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\Delta t_X$ (s)</td>
<td>100</td>
<td>900</td>
</tr>
<tr>
<td>$\phi_{A_D}$ (m)</td>
<td>2.0</td>
<td>20</td>
</tr>
<tr>
<td>$\phi_{A_I}$ (m)</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>$\phi_{R_D}$</td>
<td>0.50</td>
<td>0.75</td>
</tr>
<tr>
<td>$\phi_{R_I}$</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

7.6.1 Example 1 – Adaptive Modeling

Example 1 concerns the small looped network from Onizuka (1986; Figure 7.3) with a tank diameter of 1 m and pipe wave speeds of 1,000 m/s. Initially, the system is at steady state with both valves open, and unsteady flow conditions are induced by their simultaneous closure. To validate the comprehensive AHTM, it is compared against an MOC-based water hammer model for two events – sudden valve closure and uniform valve closure over 100 s. Simulation results for the head at node J2 are plotted in Figure 7.7.

Figure 7.7 shows that the AHTM and GCM simulation results compare well. For sudden valve closure, the AHTM implements the CGGA as the GCM to capture the resulting highly unsteady flow conditions, though fluid cavitation, which would otherwise arise from the extreme negative pressures, is neglected. The AHTM simulated the slow valve closure event using a combination of each type of model. Indeed, Figure 7.7b shows that the results agree up until dynamic effects become negligible; after this, there are minor residual inertial oscillations due to the presence of the tank.

Altogether, this example illustrates two key features of the current work. First, the comprehensive AHTM adapts to the degree of unsteadiness, and second, appropriate indicator tolerances are necessary to obtain physically representative results. Practically, the tolerances can be chosen according to what the analyst considers acceptable physical link imbalance error. If smaller values are used, the AHTM will implement more water hammer and RWC microsimulations, though at the expense of greater computational effort. Indicator tolerances can also be prescribed depending on which events are of interest, a key part of example 2 below.
Chapter 7: Comprehensive Adaptive Hybrid Modeling

(a) Event 1 – sudden valve closure

(b) Event 2 – valve closure over 100 s

Figure 7.7: Example 1 simulation results for node J2
7.6.2 Example 2 – Power Failure and Recovery in an EPS

Analyses of WDSs frequently rely on EPSs. They are performed for various purposes, for example, to evaluate system capacity, optimize operations, and evaluate structural integrity. Example 2 performs a 24 h EPS for the water transmission network from Nault and Karney (2016b; Figure 7.8) to contrast the different types of unsteady flow models; this analysis additionally considers an intermediate global power failure at all source pumping stations. Power failures are often the most problematic event for WDSs, often resulting in extreme pressures that may collapse weak pipes or allow contaminant intrusion. It is thus of interest to consider the response and recovery from such events.

Figure 7.8: Pipe network 2 – water transmission network (Nault and Karney, 2016b)

Apart from the power failure event, the same operations are performed as listed in Nault and Karney (2016b). The power failure event is simulated during peak system demand at 10:30 am, after which the failed pumps are restarted by staggered 15 minute intervals at each station. A water hammer model can simulate the transient event, but it would be impractical to model hydraulics beyond this period. Results for the current AHTM, that of Nault and Karney (2016b) using the HGGA (with $\Delta t_{RWC}$ and $\Delta t_x$), and a quasi-steady model (using $\Delta t_x$) are shown in Figure 7.9 through Figure 7.12.
Chapter 7: Comprehensive Adaptive Hybrid Modeling

Figure 7.9: Example 2 EPS results for Zone 1 Pump Station 1 discharge head

Figure 7.10: Example 2 EPS results for Zone 1 Pump Station 2 discharge head
Figure 7.11: Example 2 EPS results for Zone 1 Pump Station 4 discharge head

Figure 7.12: Example 2 EPS results for Zone 1 Pump Station 5 discharge head
Chapter 7: Comprehensive Adaptive Hybrid Modeling

From Figure 7.9 through Figure 7.12, each model predicts the same long-term behaviour. The results immediately following each operation, however, differ moderately to significantly. Dynamic effects are not captured by the quasi-steady model, which uses larger $\Delta t$ and neglects compressibility and inertial effects; its results also plot ahead of the other models’. Comparatively, the AHTMs largely agree except during the emergency pump stop event when compressibility effects are clearly important (Figure 7.13 through Figure 7.16). The HGGA-based AHTM does not accurately capture the resulting transient hydraulics, yet the comprehensive AHTM does. It even compares well against results from a water hammer model. Moreover, the comprehensive AHTM simulates the system’s unsteady flow conditions until dynamic effects dissipate, including the residual inertial oscillations arising from surge vessels at the source pumping stations.

In addition to physical accuracy, computational efficiency is essential for analyzing the unsteady hydraulics of large networks. The efficiency of the AHTM depends on multiple factors, including the solution parameters (Table 7.2), network properties, and event characteristics. These all vary between analyses and systems, so it is impractical to generalize computational performance. Instead and for illustrative purposes, Table 7.3 compares the number of time steps implemented by each model. Indeed, those for the comprehensive AHTM are significantly higher, mostly due to the water hammer microsimulations following the power failure event; if computational efficiency were more important than physical accuracy, less stringent indicator tolerances could be used, though at the expense of reduced physical accuracy. This is characteristic of the trade-off between computational efficiency and physical accuracy.

<table>
<thead>
<tr>
<th>Solution Step</th>
<th>AHTM with CGGA</th>
<th>AHTM with HGGA</th>
<th>Water Hammer Model</th>
<th>RWC Model</th>
<th>Quasi-Steady Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water Hammer</td>
<td>11,700</td>
<td>-</td>
<td>1,730,000</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RWC</td>
<td>2,490</td>
<td>2,970</td>
<td>-</td>
<td>86,400</td>
<td>-</td>
</tr>
<tr>
<td>Quasi-Steady</td>
<td>97</td>
<td>97</td>
<td>-</td>
<td>-</td>
<td>97</td>
</tr>
<tr>
<td>Total</td>
<td>14,290</td>
<td>3,070</td>
<td>1,730,000</td>
<td>86,400</td>
<td>97</td>
</tr>
</tbody>
</table>
Figure 7.13: Example 2 power failure results for Zone 1 Pump Station 1 discharge head

Figure 7.14: Example 2 power failure results for Zone 1 Pump Station 2 discharge head
Chapter 7: Comprehensive Adaptive Hybrid Modeling

Figure 7.15: Example 2 power failure results for Zone 1 Pump Station 4 discharge head

Figure 7.16: Example 2 power failure results for Zone 1 Pump Station 5 discharge head
7.6.3 Practical Remarks

In addition to the examples, there are two topics worth discussing further. First is the matter of selecting suitable indicator tolerances, which may initially appear arbitrary. The second topic concerns practical unsteady flow modeling; despite being more involving than the conventional approach, there are certainly benefits to considering a comprehensive AHTM.

Derived from the governing equations, $\phi_A$ and $\phi_R$ actively guide model selection for the comprehensive AHTM. Key to this are the indicator tolerances, and they must be chosen carefully. If the tolerances are too loose, then the AHTM may not perform as intended; conversely, values too tight will result in computationally demanding simulations. Selecting suitable tolerances thus poses a challenge. Because $\phi_A$ and $\phi_R$ approximate the physical error of incompressible flow models, one approach is to specify tolerances based on acceptable levels of error (Nault and Karney, 2016b). For example, given maximum link head difference errors of 0.1 m and 5% for a quasi-steady model, one could take $\phi_{\text{A}i} = 0.1$ m and $\phi_{\text{R}i} = 0.05$, as was done for example 1. This is less pragmatic for larger networks though. Instead, and more broadly, tolerances can be adopted according to the purpose of the analysis. In example 2, larger dynamic tolerances were expressly chosen for the power failure event to minimize additional water hammer microsimulations. Sensitivity tests can even be performed to assess how sensitive the tolerances are. Rather than a limitation, the ability to adapt to different analyses is an advantage.

Although flexible, the present AHTM is admittedly more complex than conventional modeling. An apt understanding of unsteady flow simulation is needed, particularly with respect to the preceding discussion. Modeling dynamic events also requires detailed information about their implementation, and this goes beyond the standard data provisions for EPSs. Moreover, quasi-steady models are well-suited to performing EPSs and water hammer models to extreme events. Justifying a comprehensive AHTM pertains to its generality. Certainly, it is simpler to employ a single type of model given a single analysis objective, yet a comprehensive model generalized to multiple objectives is more powerful. The present AHTM can perform EPSs, analyses of extreme conditions, and combinations of both, the latter of which was demonstrated in example 2. Other models, including previous AHTMs, do not perform as well due to either numerical stability concerns (Axworthy, 1997), significant computational demand (Filion and Karney, 2002), or a lack of physical accuracy (Nault and Karney, 2016b). Together with its ability to balance the accuracy-efficiency tension, the central benefit of the present comprehensive AHTM is its generality.
7.7 Conclusions

Computational models are essential for analyzing unsteady pipe network hydraulics. Different physical assumptions lead to different types of models, so there is a trade-off between computational efficiency and physical accuracy. To balance these competing demands, an AHTM is presented that combines the CGGA with unsteady flow characterization and an adaptive scheme. The CGGA generalizes the HGGA from Chapter 5 (see also Nault and Karney, 2016b) and the GCM from Chapter 6 to a single unified framework, a significant development that allows the formulation to implement either or a combination of these solvers. Together with unsteady flow indicators, the adaptive scheme actively adjusts the CGGA and $\Delta t$ according to the degree of unsteadiness; in this way, dynamic effects are only considered when necessary to perform efficient simulations.

Compared to previous AHTMs, the current work is stable, physically accurate, and computationally efficient. To demonstrate these features, the comprehensive AHTM was applied to two examples, a small looped network and the EPS of a water transmission system. Key to the AHTM’s successful application are its generality and suitable unsteady flow indicator tolerances. Though the latter must be chosen carefully, the tolerances can be specified to obtain physically valid results, or they may represent an acceptable level of physical error for particular events. Certainly the ability to adapt to individual analyses is both advantageous and significant. Altogether, while any model is in fact that, a model of reality, a general purpose adaptive model is a powerful tool.
Key Notation

\( \Delta t \) Time step (s)
\( \Delta t_x \) Extended time step (s)
\( \Delta t_{RWC} \) RWC time step (s)
\( \Delta t_{WH} \) Water hammer time step (s)
\( \varepsilon \) Friction integration parameter
\( \theta \) Momentum integration parameter
\( \theta \) Type 1 node mass integration parameter
\( A_0 \) \( \mathbb{R}^{n_0 \times n_L} \) Type 1 node-link incidence matrix
\( A_{0A} \) \( \mathbb{R}^{n_0 \times n_L} \) Type 0 link start node incidence matrix
\( A_{0B} \) \( \mathbb{R}^{n_0 \times n_L} \) Type 0 link end node incidence matrix
\( A_1 \) \( \mathbb{R}^{n_1 \times n_L} \) Type 1 node-link incidence matrix
\( A_{1A} \) \( \mathbb{R}^{n_1 \times n_L} \) Type 1 node-start link incidence matrix
\( A_{1B} \) \( \mathbb{R}^{n_1 \times n_L} \) Type 1 node-end link incidence matrix
\( B \) \( \mathbb{R}^{n_L} \) Link compressibility effects constant vector (s/m²)
\( B_A(*) \) \( \mathbb{R}^{n_L \times n_L} \) Link start flow dynamic effects vector operator (m)
\( B_B(*) \) \( \mathbb{R}^{n_L \times n_L} \) Link end flow dynamic effects vector operator (m)
\( b_A \) \( \mathbb{R}^{n_L} \) Link start flow reach-back vector (m)
\( b_B \) \( \mathbb{R}^{n_L} \) Link end flow reach-back vector (m)
\( C_1 \) \( \mathbb{R}^{n_1 \times n_1} \) Diagonal type 1 node area matrix (m²)
\( F(*) \) \( \mathbb{R}^{n_L} \) Incompressible flow head loss vector operator (m)
\( G_1(*) \) \( \mathbb{R}^{n_1} \) Type 1 node mass balance vector (m³/s)
\( H_0 \) \( \mathbb{R}^{n_0} \) Type 0 node head vector (m)
\( H_1 \) \( \mathbb{R}^{n_1} \) Type 1 node head vector (m)
\( J_{QF} \) Quasi-steady friction unit head loss
\( J_{UF} \) Unsteady friction unit head loss
\( L_I \) \( \mathbb{R}^{n_L} \) Link inertia coefficient vector (s²/m²)
\( m \) Previous iteration index
\( n_0 \) Number of type 0 nodes
\( n_1 \) Number of type 1 nodes
Chapter 7: Comprehensive Adaptive Hybrid Modeling

\begin{align*}
  n_{ICF} & \quad \text{Number of independent compressible flows} \\
  n_{DT1} & \quad \text{Number of dependent type 1 nodes} \\
  n_{IF} & \quad \text{Number of incompressible flows} \\
  n_{IT1} & \quad \text{Number of independent type 1 nodes} \\
  n_L & \quad \text{Number of links} \\
  n_N & \quad \text{Number of nodes} \\
  n_{NCF} & \quad \text{Number of non-linearly-related compressible flows} \\
  P_1 & \quad \mathbb{R}^{n_1 \times n_1} \quad \text{Type 1 node permutation matrix} \\
  P_A & \quad \mathbb{R}^{n_L \times n_L} \quad \text{Link start flow permutation matrix} \\
  P_B & \quad \mathbb{R}^{n_L \times n_L} \quad \text{Link end flow permutation matrix} \\
  Q_A & \quad \mathbb{R}^{n_L} \quad \text{Start flow vector (m}^3/\text{s}) \\
  Q_B & \quad \mathbb{R}^{n_L} \quad \text{End flow vector (m}^3/\text{s}) \\
  q_1 & \quad \mathbb{R}^{n_1} \quad \text{Type 1 node outflow vector (m}^3/\text{s}) \\
  R_A & \quad \mathbb{R}^{n_L} \quad \text{Right hand side vector (m)} \\
  R_B & \quad \mathbb{R}^{n_L} \quad \text{Right hand side vector (m)} \\
  R_I & \quad \mathbb{R}^{n_L} \quad \text{Right hand side vector (m)} \\
  S_C & \quad \mathbb{R}^{n_L \times n_L} \quad \text{Binary link compressibility solver matrix} \\
  S_I & \quad \mathbb{R}^{n_L \times n_L} \quad \text{Binary link inertial solver matrix}
\end{align*}
Chapter 8: Summary and Key Findings

From municipal water supply systems to parts of wastewater collection networks, many urban water systems comprise pressurized pipe networks. Essential to analyzing the unsteady hydraulics of such systems are models. In addition to networks with intricate topologies, modeling concerns an array of unsteady flow phenomena, including fluid cavitation and column separation, pressure wave interaction, and unsteady energy dissipation; moreover, there is the matter of model selection. Multiple types of models exist, each featuring different physical accuracy and computational efficiency (Figure 8.1). The more efficient quasi-steady models are valid for slow varying flow conditions, whereas highly transient conditions require more physically accurate yet computationally demanding RWC or water hammer models. Accordingly, there is an accuracy-efficiency trade-off. To balance these competing demands, this thesis develops a generalized adaptive hybrid transient model (AHTM) capable of simulating the full range of one-dimensional unsteady pipe network hydraulics.

Figure 8.1: Conceptual relationship between the different types of models

In developing the comprehensive AHTM, gaps in the literature were addressed sequentially. Chapter 4 first treated RWC models. Existing RWC formulations suffer numerical instabilities, even for modestly-sized networks: this predominantly results from using explicit numerical methods to solve a system of stiff ordinary differential equations. To resolve this, Chapter 4 presents two improved RWC formulations for simulating unsteady-incompressible pipe network flows. The first, termed the RWC global gradient algorithm (GGA), extends the quasi-steady
Chapter 8: Conclusions and Key Findings

formulation of Giustolisi et al. (2012) to consider inertial effects, and the second formulation combines Shimada's (1989) incidence method with implicit numerical integration to improve numerical stability. Owing to its similarity to the GGA-like solvers (Todini and Pilati, 1988; Todini, 2011; Giustolisi et al., 2012; Todini and Rossman, 2013), the RWC GGA features favourable numerical traits and benefits from the relevant literature. Both RWC formulations are shown to overcome the numerical challenges of previous work, with the RWC GGA being more efficient. Table 8.1 summarizes the key differences between the RWC formulations presented in Chapter 4 against previous work.

**Table 8.1: Comparison of current RWC formulations against previous work**

<table>
<thead>
<tr>
<th>RWC Formulation</th>
<th>Source</th>
<th>Loops</th>
<th>Nodal Continuity</th>
<th>Head Loss Treatment</th>
<th>Numerical Integration</th>
<th>Numerical Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWC GGA</td>
<td>Chapter 4</td>
<td>No</td>
<td>Implicit</td>
<td>Standard</td>
<td>Variable</td>
<td>Stable</td>
</tr>
<tr>
<td>Implicit Incidence Method</td>
<td>Chapter 4</td>
<td>No</td>
<td>Implicit</td>
<td>Standard</td>
<td>Variable</td>
<td>Stable</td>
</tr>
<tr>
<td>Gradient Algorithm</td>
<td>Ahmad (1997)</td>
<td>No</td>
<td>Implicit</td>
<td>Linearized</td>
<td>Fixed</td>
<td>Stable</td>
</tr>
<tr>
<td>Loop-Based Methods</td>
<td>Holloway (1985), Onizuka (1986)</td>
<td>Yes</td>
<td>Implicit</td>
<td>Standard</td>
<td>Fixed</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

Unlike RWC models, quasi-steady models are well-suited to extended period simulations (EPSs). Dynamic effects are presumed negligible in such analyses, but this is seldom the case in reality. Chapter 5 introduces an incompressible flow AHTM to improve physical accuracy for EPSs. The AHTM combines a hybrid solver, termed the hybrid GGA (HGGA), with unsteady flow characterization and an adaptive scheme; together, the adaptive scheme adjusts the HGGA and time step according to unsteady flow indicators and their tolerances. In this way, inertial effects are only considered when necessary to permit efficient yet physically accurate simulations (subject to the physical validity of RWC models). From comparisons against a water hammer model, the AHTM was found to perform well, even yielding comparable results for routine operations in water distribution systems. Further, the unsteady flow indicator tolerances can be specified according to the purpose of the analysis. Indeed the incompressible flow AHTM is a
powerful tool, but it features a key limitation: it cannot simulate unsteady-compressible flow.

Network hydraulics are highly transient under unsteady-compressible flow. Of the three types of models (Figure 8.1), only water hammer models are applicable, for they consider friction, inertial, and compressibility effects. Historically, most water hammer models have relied on the method of characteristics (MOC; Wylie and Streeter, 1978; Chaudhry, 2014). Though proven and easily implemented, the MOC is more computationally demanding than two of its alternatives: the wave characteristics method (WCM; Wood et al., 2005) and algebraic water hammer (AWH; Wylie and Streeter, 1993). To benefit from the MOC’s greater spatial resolution and the greater computational efficiency of the WCM and AWH, Chapter 6 generalizes these characteristic type solution methods to the generalized characteristic method (GCM). Essentially, the GCM combines a flexible friction integral approximation with a variable reach scheme. By allowing variable order representations of individual pipes, the solution method can be adapted according to the required degree of accuracy; consequently, more accurate solutions are only provided where necessary to facilitate efficient simulations. Three examples demonstrate the GCM. In addition to the GCM yielding similar results to the MOC, comparisons of the methods illustrates how the GCM requires significantly fewer computations than even the WCM. Table 8.2 contrasts the key differences between the GCM against the MOC, WCM, and AWH.

Table 8.2: Comparison of current water hammer formulations against previous work

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Source</th>
<th>Friction Integral Approximation</th>
<th>Interior Reaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCM</td>
<td>Chapter 6</td>
<td>Variable Implicit*</td>
<td>1 to 2 $N_0$ †</td>
</tr>
<tr>
<td>MOC</td>
<td>Wylie and Streeter (1978); Chaudhry (2014)</td>
<td>First Order Explicit*</td>
<td>$N_0$</td>
</tr>
<tr>
<td>WCM</td>
<td>Wood et al. (1966)</td>
<td>First Order Implicit*</td>
<td>2</td>
</tr>
<tr>
<td>AWH</td>
<td>Wylie and Streeter (1978)</td>
<td>First Order Explicit</td>
<td>1</td>
</tr>
</tbody>
</table>

* Though the MOC has been used with second order explicit schemes, it is commonly used with a first order explicit scheme
† $N_0$ is the base number of discretized interior reaches for a pipe.

Chapters 4, 5, and 6 treat incompressible and unsteady-compressible network flows separately. The significance of these chapters is as follows:

- A novel RWC formulation, the RWC GGA, is introduced in Chapter 4 that overcomes the numerical challenges of previous work. Put simply, Chapter 4 shows that the RWC
GGA can be obtained via the addition of a few terms to the G-GGA.

- Significantly, Chapter 5 generalizes the G-GGA and RWC GGA to a single formulation, the HGGA. An AHTM based on the HGGA is developed to perform EPSs efficiently yet with greater physical accuracy than conventional EPSs: this is accomplished by combining the HGGA with unsteady flow characterization and an adaptive scheme.

- Similar to the HGGA for incompressible network flow analysis, Chapter 6 generalizes the MOC, WCM, and AWH to a single formulation, the GCM. The characteristic-based methods are shown to be fundamentally similar; moreover, the GCM allows greater flexibility, numerical accuracy, and computational efficiency than any single one of the aforementioned formulations.

Chapter 7 represents the culmination of this thesis. A comprehensive AHTM is developed, one capable of simulating the full range of unsteady pipe network hydraulics. Like the incompressible flow AHTM of Chapter 5, the comprehensive AHTM comprises three components: a hybrid solver, unsteady flow characterization indices, and an adaptive scheme. The solver, termed the comprehensive global gradient algorithm (CGGA), generalizes the HGGA and GCM, so it can represent individual pipes using any of a water hammer, RWC, or quasi-steady model. Accordingly, the CGGA relates relevant water hammer, RWC, and quasi-steady modeling formulations to a single unified framework – this is key to the current work transitioning from implementing one type of model to another. Essential to determining how to select which model to use are unsteady flow characterization indices and their tolerances. As an extension of the work in Chapter 5, two indices are proposed. The CGGA and indices are combined with an adaptive scheme that adjusts the solver and time step according to the degree of unsteadiness; in this way, more computationally demanding models are only used when their greater physical accuracy is warranted. Compared to Filion and Karney (2002), the present comprehensive AHTM represents a more flexible approach, and unlike Axworthy (1997), numerical stability is not problematic. Further, using the GCM instead of the MOC permits efficient simulations even when the AHTM implements water hammer solves. Altogether, the present AHTM is comprehensive in nature; it relates water hammer, RWC, and quasi-steady type models within a common framework; the previously noted limitations (both numerical and physical) are overcome; and the comprehensive AHTM is flexible in its implementation as either or a combination of the aforementioned. To summarize these significant developments, Table 8.3 contrasts the present AHTMs against the

### Table 8.3: Comparison of present adaptive hybrid transient models against others

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Quasi-Steady Solver</td>
<td>CGGA (Chapter 7) as the G-GGA (Giustolisi et al., 2012)</td>
<td>HGG (Chapter 5) as the G-GGA (Giustolisi et al., 2012)</td>
<td>Incidence Method (Shimada, 1989 with implicit integration)</td>
<td>GGA (Todini and Pilati, 1988)</td>
</tr>
<tr>
<td>RWC Solver</td>
<td>CGGA (Chapter 7) as the RWC GGA (Chapter 4)</td>
<td>HGG (Chapter 5) as the RWC GGA (Chapter 4)</td>
<td>Incidence Method (Shimada, 1989 with explicit integration)</td>
<td>None</td>
</tr>
<tr>
<td>Water Hammer Solver</td>
<td>CGGA (Chapter 7) as the GCM (Chapter 6)</td>
<td>None</td>
<td>MOC (Wylie and Streeter, 1978; Chaudhry, 2014)</td>
<td>MOC (Wylie and Streeter, 1978; Chaudhry, 2014)</td>
</tr>
<tr>
<td>Adaptive Scheme</td>
<td>Event-oriented with physically-based transition indices and variable-duration microsimulations</td>
<td>Event-oriented with physically-based transition indices and variable-duration microsimulations</td>
<td>Event-oriented with numerically-based transition indices and variable-duration microsimulations</td>
<td>Event-oriented with fixed-duration microsimulations</td>
</tr>
</tbody>
</table>

Indeed the comprehensive AHTM is a powerful tool and a significant development. Unlike conventional modeling with a single type of unsteady flow model, the AHTM simulates network hydraulics from steady to unsteady-compressible flow. It also seamlessly transitions from one model type to another. A general purpose model such as this features numerous advantages. It alleviates the need for model selection, and by using more physically accurate models only when necessary, unsteady network flow simulations can be performed efficiently. For design, this permits the consideration of a greater number of analysis scenarios, thus reducing the probability of overlooking critical system states; moreover, the methodology can simulate the hydraulics of larger networks without the need for gross model simplification, known as skeletonization. Despite this, the reader should be mindful of the limitations of AHTMs, just as with any model.

There are three key limitations to AHTMs. These comprise selecting indicator tolerances, model complexity, and input data. Central to the AHTMs presented here are the indicator tolerances that guide model selection. Different tolerances will yield different simulation results, so care is needed in choosing them. Because the unsteady flow indicators characterize the degree of a model’s physical inaccuracy, it is prudent to select the indicator tolerances according to an acceptable level of physical model error. This varies with analyses and the analyst’s preferences.
Additionally, no two pipe networks are alike, so sensitivity tests should be conducted to assess how simulation results vary with different indicator tolerances; in fact, a preferable approach is to perform a simulation with large tolerances, observe the computed indicator values, and then specify the tolerances for an analysis based on the events of interest. Not only does this allow model selection based on the underlying unsteady flow physics, but it also reflects the purpose of an analysis. For example, one can select larger tolerances if only extreme events are of concern. Regarding the second limitation, AHTMs are admittedly more complex than type-specific models. Nonetheless, AHTMs can be implemented as any type-specific model by judiciously selecting appropriate indicator tolerances; moreover, the key benefit of an AHTM is its ability to be implemented as any type-specific model or combination thereof. The last matter concerns input data. Certainly there are uncertainties to input data, and one may question the merits of using a more physically accurate model given input data uncertainties, particularly with respect to the design of new systems and additions thereto. In such cases, it is prudent to perform sensitivity tests, to which the current work is readily suited given its greater computational efficiency compared to conventional modeling approaches. Ultimately, AHTMs, are exactly that, a replica of reality. Accordingly, they should be treated as such.

Beyond the above there remains room for further work, most notably with respect to implementation and applications. Through using more efficient numerical models, design optimization can benefit from exploring a broader range of possible solutions. As an example, transient flow analyses typically consider designing for a system’s worst-case response: key to this is determining the conditions that yield a system’s worst-case response. Access to more efficient models also enables the exploration of broader solution domains to identify not only the most critical conditions, but also the system’s sensitivity to key parameters. Efficient modeling formulations also make real-time modeling more practical. Presently, only quasi-steady models are a viable alternative for real-time analyses of, for example, water distribution systems. Unsteady flow modeling with even RWC type models has yet to be explored, but the present work certainly improves the prospect, making it more practical to consider real-time analyses of unsteady flows where dynamic effects are important. The author envisions modifying the commercial software EPANET to do exactly this, for both EPANET (which is based on the GGA; Todini and Pilati, 1988) and the HGGA share the same solution framework. Altogether, while models such as the topic work here are just that, an approximation of reality, the ability to improve the performance of a model ultimately aids the performance of its analyses.
References


Boyd, G. R.; Wang, H.; Britton, M. D.; Howie, D. C.; Wood, D. J.; Funk, J. E.; and Friedman, M
References


References

10.1061/(ASCE)WR.1943-5452.0000431, 04014040, 140(4), 435-443.


References


Water Distribution Systems Analysis Symposium 2006, Cincinnati OH.


Appendices
### Appendices

#### Outline

**Appendix A: Review of Numerical Methods**

A1. Systems of Linear and Non-Linear Equations  181
   - A1.1 Matrix Notation and Operations  182
   - A1.2 Eigenvalues and Eigenvectors  183
   - A1.3 Direct Methods for Systems of Linear Equations  185
   - A1.4 Iterative Methods for Systems of Linear Equations  191
   - A1.5 Non-Linear Systems of Equations and the Newton Method  197
   - A1.6 Other Topics  199
   - A1.7 Summary  204

A2. Numerical Solution of Systems of ODE IVPs  205
   - A2.1 Initial Value Problems  205
   - A2.2 Common Solution Schemes  207
   - A2.3 Stiffness and Solution Stability  211
   - A2.4 Summary  215

A3. Brief Remarks on Pipe Network Analysis  215

**Appendix B: Generalized Characteristic Method Implementation Algorithms**

   - B1.1 Compatibility Expressions and Solutions  219
   - B1.2 Interior Analysis Implementation  220
   - B1.3 Junction Analysis Implementation  222

B2. GCM with Quasi-Steady Friction  224
   - B2.1 Compatibility Expressions and Solutions  224
   - B2.2 Interior Analysis Implementation  225
   - B2.3 Junction Analysis Implementation  226

B3. GCM with Steady Friction  228
   - B3.1 Compatibility Expressions and Solutions  228
   - B3.2 Interior Analysis Implementation  229
   - B3.3 Junction Analysis Implementation  230

B4. GCM with Steady Friction and $\epsilon = 0$  232
   - B4.1 Compatibility Expressions and Solutions  232
Appendices

B4.2 Interior Analysis Implementation .................................................. 233
B4.3 Junction Analysis Implementation ..................................................... 233

B5. GCM with Steady Friction, $\theta = 0.5$, and $\epsilon = 0$ ......................... 235
B5.1 Compatibility Expressions and Solutions ........................................... 235
B5.2 Interior Analysis Implementation ...................................................... 236
B5.3 Junction Analysis Implementation ..................................................... 237

B6. GCM with Steady Friction, $\theta = 1$, and $\epsilon = 0$ .......................... 239
B6.1 Compatibility Expressions and Solutions ........................................... 239
B6.2 Interior Analysis Implementation ...................................................... 239
B6.3 Junction Analysis Implementation ..................................................... 240

B7. GCM with Steady Friction and $\theta = 0$ ............................................ 242
B7.1 Compatibility Expressions and Solutions ........................................... 242
B7.2 Interior Analysis Implementation ...................................................... 243
B7.3 Junction Analysis Implementation ..................................................... 243

B8. GCM with Steady Friction, $\theta = 0$, and $\epsilon = 0.5$ ......................... 244
B8.1 Compatibility Expressions and Solutions ........................................... 244
B8.2 Interior Analysis Implementation ...................................................... 245
B8.3 Junction Analysis Implementation ..................................................... 245

B9. GCM with Steady Friction, $\theta = 0$, and $\epsilon = 1$ .......................... 247
B9.1 Compatibility Expressions and Solutions ........................................... 247
B9.2 Interior Analysis Implementation ...................................................... 247
B9.3 Junction Analysis Implementation ..................................................... 248

B10. GCM with Steady Friction and $\theta = \epsilon = 0$ .................................. 249
B10.1 Compatibility Expressions and Solutions ........................................... 249
B10.2 Interior Analysis Implementation ...................................................... 249
B10.3 Junction Analysis Implementation ..................................................... 250
Appendix A: Review of Numerical Methods

Many numerical problems involve interdependent equations that share a common set of variables. To facilitate their solution, such problems are arranged into a system of equations. Systems of equations and systems of differential equations range in complexity, dimension, and linearity depending on the problem, and their solution is key to many scientific and engineering models, including the topic of this thesis. In solving these, two areas of numerical analysis play key roles: linear algebra for solving systems of equations, and numerical integration methods for solving systems of differential equations.

This appendix reviews concepts that are fundamental to the aforementioned topics. Numerical solutions to systems of equations via direct and iterative methods are treated first in §A1, and then methods for solving systems of linear ODE IVPs are summarized in §A2. The material is predominantly based on Strang (1986a and 1986b), Benzi et al. (2005), and Epperson (2002), each of whom treats one or more of the topics in greater detail. Indeed, these and numerous other mathematical texts cover numerical analyses. However, the subject matter is vast and widespread, so this appendix provides a helpful overview. It is not the objective to provide a comprehensive mathematical treatment of numerical methods; rather, common concepts are introduced to provide sufficient background. As such, proofs and formality are omitted. If interested, the reader is directed to the references for further details.

Throughout this appendix, both scalar and vector terms are used. Notation for the former uses italicized characters (e.g., $a$), while vector and matrix terms are represented using italicized bold-faced uppercase and lowercase characters, respectively (e.g., $A$ and $x$). Further, the $(i,j)$-th element of a matrix $A$ is denoted by $A_{ij}$, and the m-th form of matrix $A$ is represented by $A^{(m)}$: $m$ will often denote an iteration or time step counter. The dimension of each non-scalar term is also indicated upon introduction (e.g., $\mathbb{R}^{n \times n}$).

A1. Systems of Linear and Non-Linear Equations

Systems of equations contain exactly what their name implies, a collection of expressions interrelated by a set of common variables. Given a system of equations, the solution $x \in \mathbb{R}^n$ is sought such that $Ax = b$, where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. One solution approach is to compute $A^{-1}$, the inverse of $A$. The solution is then $x = A^{-1}b$. This is conceptually straightforward, so
why bother with numerical methods? Often it is too costly to compute $A^{-1}$, which itself involves solving the linear system $AX = I$. Alternatively, the system of equations can be left as $Ax = b$, and $x$ can be determined numerically.

Numerical methods for solving systems of *linear* and *non-linear* equations are explored below. Fundamental concepts are reviewed first, including matrix notation, common operations, and key properties. Then, we continue with direct and iterative approaches: the former are suitable for small and dense systems, while the latter are more appropriate for large and sparse cases. These are subsequently extended to the solution of *non-linear systems equations* via the Newton method. Further topics are discussed in relation to the aforementioned, including saddle point problems (a particular type of structure system of equations) and sparse systems of equations.

**A1.1 Matrix Notation and Operations**

It is helpful to begin with a review of fundamental concepts. A $C^{n \times m}$ system of equations comprises a collection of $m$ expressions with $n$ variables, such as that shown below:

$$
\begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
A_{m1} & A_{m2} & \cdots & A_{mn}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix}
=
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{bmatrix}
$$

[A1]

where $A_{ij} \in C = a$ coefficient; $x_j \in C = a$ variable; $b_i \in C = a$ constant; $i =$ the row index $(i = [1, m])$; and $j =$ the column index $(j = [1, n])$. Herein, only $A_{ij}$, $x_j$, and $b_i \in R$ are considered, and we restrict ourselves to cases involving square ($n \times n$) coefficient matrices. Such problems usually have a unique solution, while others (where $m \neq n$) do not.

Mathematically, it is more convenient to write Equation [A1] as

$$Ax = b$$

[A2]

where $A \in R^{n \times n} = a$ coefficient matrix, $x \in R^n = a$ solution vector, and $b \in R^n = a$ constant vector. If $A$ is constant and the exponent on $x$ is unity, then the system of equations is *linear* (meaning the elements of $x$ are *linearly* related). For systems of *non-linear* equations, $A = A(x) \in R^{n \times n}$ is a function of $x$, where $A(*) : R^n \to R^n = a$ operator that transforms an $R^n$ vector into an $R^n$ product. Systems of non-linear equations are often written as

$$Ax - b = 0$$

[A3]

where the right hand side is an $R^{n \times n}$ zero vector. It is important to distinguish between systems
Appendix A: Review of Numerical Methods

of linear and non-linear equations, for the two are solved through different (albeit related) means. Because the discussions in this and the majority of the following sections concern the linear case, Equation [A2] is used as the reference format herein.

To solve Equation [A2], a vector \( x \) is sought whose values simultaneously satisfy each of the \( n \) equations. There are multiple approaches to determining \( x \), some direct and others iterative: §A1.3 and §A1.4 cover these. In solving for \( x \), the following operations and properties are central:

- **Dot product:** for \( x \) and \( y \in \mathbb{R}^n \), the dot product \( (x, y) = x \cdot y = x^T y = \sum x_i y_i \) is a scalar. Note that \( x \cdot y^T \) is not a scalar but rather a matrix.
- **Elementary row operations:** the rows of a system of equations (Equation [A2]) can be added and subtracted to any other row, including scalar multiples thereof.
- **Multiplication:** given \( A \in \mathbb{R}^{m \times n} \) and \( B \in \mathbb{R}^{n \times r} \), the product \( x = AB \in \mathbb{R}^{m \times r} \) has elements \( x_{ik} = \sum_{j=1}^r A_{ij} B_{jk} \). An important relation is \( AB = BA \) only if \( A \) and \( B \) are both symmetric or \( A = B \).
- **Inverse:** for non-singular \( A \) and \( B \in \mathbb{R}^{n \times n} \), \( B = A^{-1} \) is the inverse of \( A \) if \( BA = I \), where \( I \in \mathbb{R}^{n \times n} \) is an identity matrix. If \( A \) is singular, it has no inverse and \( \det(A) = 0 \).
- **Transpose:** the transpose of \( A \in \mathbb{R}^{m \times n} \) is \( B = A^T \in \mathbb{R}^{n \times m} \) where \( B_{ij} = A_{ji} \).
- **Kernel:** also called the null space, the kernel of \( A \in \mathbb{R}^{m \times n} \) is denoted by \( \ker(A) \). It represents the collection of vectors within the matrix \( x \in \mathbb{R}^{n \times n} \) that satisfy \( Ax = 0 \), where every column vector of \( x \) is in the null space of \( A \).

**A1.2 Eigenvalues and Eigenvectors**

Numerical analyses frequently involve computing a matrix-vector product. That is, a coefficient matrix \( A \in \mathbb{R}^{n \times n} \) is applied to a vector \( x \in \mathbb{R}^n \); equally, \( A \) can be applied to matrices. The transformative action of \( A \) is characterized by its **eigenvectors** and **eigenvalues**, collectively referred to as **eigenpairs**. Because the eigenpairs describe \( A \), they also describe and make more obvious the stability characteristics of many numerical methods for solving systems of equations (and as discussed in §A2, systems of differential equations); thus, a brief review is helpful.

A vector \( v^{(i)} \in \mathbb{R}^n \) that satisfies \( Av^{(i)} = \lambda_i v^{(i)} \) constitutes an **eigenvector** of \( A \), where \( \lambda_i \) is an **eigenvalue**. There are in fact \( n \) sets of eigenvectors to \( A \in \mathbb{R}^{n \times n} \), each with a corresponding eigenvalue. Let \( E_\lambda \in \mathbb{R}^{n \times n} \) represent a matrix whose columns are the eigenvectors of \( A \), and denote \( \sigma(A) \in \mathbb{R}^n \) as the spectrum of \( A \), which comprises the set of corresponding eigenvalues.
The largest eigenvalue, $\rho(A) = \max|\sigma|$, is also referred to as the spectral radius.

Eigenpairs feature a number of interesting properties. By definition, the eigenvector $v_i$ retains its direction when transformed by $A$, but its magnitude is scaled by $\lambda_i$. A similar concept holds when $A$ is applied to a non-eigenvector. The transformation $y = Ax \in \mathbb{R}^n$ for some $x \in \mathbb{R}^n$ can be described by the eigenpairs of $A$. Specifically, the components of $x$ in the direction of the eigenvectors of $A$ are scaled according to their corresponding eigenvalues: in this way, the eigenvectors of $A$ can be thought of as characteristic directions (this is the origin of the German term “eigen”). From this, it follows that if all of the eigenvalues of $A$ are greater than unity, then the repeated application of $A$ to $x$ will cause it to continue to grow in magnitude. Conversely, if all of the eigenvalues are less than unity (i.e., $\rho(A) < 1$), the opposite will occur: this is important for stationary methods where a matrix is repeatedly applied to minimize the residual error of an iterative solution for systems of linear equations.

Because the difference $A - \lambda_i I$ is a singular matrix, the eigenvalues of $A$ can be determined as any $\lambda_i$ that satisfy

$$\det(A - \lambda I) = 0$$

[A4]

This process is referred to as solving for the roots of the system’s characteristic polynomial. For $\mathbb{R}^{n \times n}$ matrices where $n \leq 5$, the eigenvalues can be computed explicitly; otherwise, numerical methods are needed (this is known as the eigenvalue problem). The characteristics of a matrix may also hint at or even yield its eigenvalues. For triangular matrices (which include diagonal matrices), their eigenvalues are the diagonal elements. If a matrix is symmetric ($A = A^T$), then all of its eigenvalues are real numbers. Furthermore, the eigenvalues of positive-definite matrices (which are symmetric) are all positive real numbers, and those of negative-definite matrices are all negative real numbers (definiteness is discussed in §A1.3.3).

Further to the above, the eigenvalues of a matrix characterize its condition number $\kappa(A)$. Matrices whose eigenvalues range significantly in magnitude are ill conditioned, that is, near singular, and they have $\kappa(A) \gg 1$. Singular matrices are defined by $\kappa(A) = \infty$. An important relation for normal real matrices (defined by $AA^T = A^TA$, which encompass symmetric matrices) is that their condition number is $\kappa(A) = \max|\sigma(A)| / \min|\sigma(A)|$.

Throughout the remainder of this appendix, eigenpairs play key roles in various numerical methods. The brief discussion here is largely limited to systems of equations, but eigenpairs also arise in the solution of systems of differential equations, specifically, the solution of so-called stiff
systems of differential equations (see §A2.3).

### A1.3 Direct Methods for Systems of Linear Equations

Numerical approaches to solving systems of linear equations can be categorized as either *direct* or *iterative methods*. Direct methods employ Gaussian elimination (or its variants) to compute the solution, while iterative methods approximate the solution by minimizing the residual error. Both feature favourable characteristics for different problems. Direct methods are suitable for small and dense systems of equations, and large sparse systems are more efficiently solved via iterative methods. It helps to be familiar with each, for concepts such as preconditioning are based on both approaches. Direct methods are summarized first.

#### A1.3.1 Gaussian Elimination

All direct methods are an extension of *Gaussian elimination*. As a refresher, Gaussian elimination uses elementary row operations to reduce a system of equations $[A \mid b]$ to one where the coefficient matrix is the identity matrix, i.e., $[I \mid b']$; the solution is then $x = b'$. An easy approach to automating this is to reduce $A$ to a diagonal matrix by using its main diagonal elements to eliminate the off-diagonal elements. An algorithm for this is shown below:

```matlab
% Part A – reduce A to diagonal form:
For j = 1: n  % Loop through column 1 to column n.
    For i = 1: n  % Loop through row 1 to row n.
        If i ≠ j
            m = A_{ij}/A_{jj}
            For k = 1: n  % Loop through column 1 to n for row i.
                A_{ik} = A_{ik} - m * A_{jk}
            End
            b_i = b_i - m * b_j
        End
    End
End

% Part B – compute the solution.
```
Appendix A: Review of Numerical Methods

For $j = 1: n$
\[ x_j = b_j/A_{jj} \]
End

Another direct approach is naïve Gaussian elimination. Like Gaussian elimination, the main diagonal elements are used to eliminate the off-diagonals, but $A$ is instead reduced to upper triangular form rather than diagonal form. Triangular matrices are easily solved by working along their diagonals, starting from the row with the most zeroes. Naïve Gaussian elimination is typically computationally faster than Gaussian elimination, for the system is only reduced to triangular rather than diagonal form. Sample implementation code is shown below:

% Part A – factorize $A$ into an upper triangular matrix.
For $j = 1: n - 1$  % Loop through column 1 to column $n - 1$.
    For $i = j + 1: n$  % Loop through row $j + 1$ to row $n$.
        $m = A_{ij}/A_{jj}$
        For $k = 1: n$  % Loop through column 1 to column $n$ for row $i$.
            $A_{ik} = A_{ik} - m * A_{ij}$
        End
    End
$\quad b_i = b_i - m * b_j$
End
End

% Part B – solve the resulting upper triangular system.
$\quad x_n = b_n/A_{nn}$  % Compute the $n$th solution.
For $i = n - 1: -1: 1$  % Loop through row $n - 1$ to row 1.
    $S = 0$  % Initialize the sum counter for each variable.
        For $j = i + 1: n$
            $S = S + A_{ij} * x_j$
        End
$\quad x_i = (b_i - S)/A_{ii}$
End
A1.3.2 Lower-Upper Decomposition

Suppose we wish to solve a system of equations multiple times for different \( \mathbf{b} \). Both Gaussian elimination and naïve Gaussian elimination require \( \mathbf{A} \) too be factored at each solve, which is rather inefficient. A better approach is to use lower-upper (LU) decomposition. This approach decomposes \( \mathbf{A} \) into two matrices, an upper triangular matrix \( \mathbf{U} \in \mathbb{R}^{n \times n} \) and a lower triangular matrix \( \mathbf{L} \in \mathbb{R}^{n \times n} \), such that \( \mathbf{A} = \mathbf{LU} \). The reduced system of equations is solved in two steps: first the triangular system \( \mathbf{L} \mathbf{y} = \mathbf{b} \) is solved for \( \mathbf{y} \in \mathbb{R}^{n} \), and then a second triangular system \( \mathbf{U} \mathbf{x} = \mathbf{y} \) is solved for \( \mathbf{x} \). If \( \mathbf{b} \) changes, the same \( \mathbf{L} \) and \( \mathbf{U} \) can be used.

Although efficient, there is now the matter of determining \( \mathbf{L} \) and \( \mathbf{U} \). Fortunately this is easily accomplished using the algorithm below:

1. Determine \( \mathbf{R}^{(k)} \in \mathbb{R}^{n \times n} \), where
   \[
   R_{ij}^{(k)} = \begin{cases} 
   A_{ij}^{(k)} / A_{jj}^{(k)} & \text{if } [k = i] \text{ and } [i > j] \\
   0 & \text{Otherwise}
   \end{cases}
   \quad [\text{A5}]
   
   for each column \( 1 \leq k < n - 1 \).

2. Determine \( \mathbf{E}^{(k)} = \mathbf{I} - \mathbf{R}^{(k)} \in \mathbb{R}^{n \times n} \) for each column \( 1 \leq k < n - 1 \), where \( \mathbf{I} \in \mathbb{R}^{n \times n} \) is an identity matrix. The elements of each \( \mathbf{E}^{(k)} \) represent the multipliers for column \( k \) needed to reduce the off-diagonals using the diagonal.

3. The upper triangular matrix is then
   \[
   \mathbf{U} = \left( \prod_{k=1}^{n-1} \mathbf{E}^{(k)} \right) \mathbf{A} = \mathbf{E}^{(n-1)} \mathbf{E}^{(n-2)} \ldots \mathbf{E}^{(2)} \mathbf{E}^{(1)} \mathbf{A} 
   \quad [\text{A6}]
   
   The nonzero elements are exactly the Gaussian elimination multipliers, and \( \mathbf{U} \) is the same upper triangular matrix as would be produced by naïve Gaussian elimination.

4. The lower triangular matrix is given as
   \[
   \mathbf{L} = \mathbf{I} + \sum_{k=1}^{n-1} \mathbf{R}^{(k)}
   \quad [\text{A7}]
   
   The matrix \( \mathbf{L} \) comprises only multipliers with main diagonal values of unity.

An implementation of the LU algorithm is provided below. Note that the respective upper and lower triangular elements or \( \mathbf{L} \) and \( \mathbf{U} \) are stored within \( \mathbf{A} \) itself to minimize the need for additional memory (Epperson, 2013).
% Part A – factorize $A$.
For $j = 1: n - 1$  % Loop through the columns of $A$.
    For $i = j + 1: n$  % Loop through row $j + 1$ to row $n$.
        $m = A_{ij}/A_{jj}$  % Calculate row multiplier.
        $A_{ij} = m$
    For $k = j + 1: n$  % Update row $i$ upper triangular elements.
        $A_{ik} = A_{ik} - m * A_{jk}$

% The same operation above could include $k = 1: i - 1$ (i.e., the below % diagonal elements of row $i$), but by definition of the row reduction % operations, these values are zero. Accordingly, the resulting % transformation is an upper triangular matrix.

End
End
End

% Part B – solve the lower triangular system $Ly = b$ for $y$.
$y_1 = b_1$  % Assign the first solution.
For $i = 2: n$  % Loop through row 2 to row $n$.
    $S = 0$  % Initialize sum counter for each variable.
    For $j = 1: i - 1$
        $S = S + A_{ij} * y_j$
    End
    $y_i = b_i - S$
End

% Part C – solve the upper triangular system $Ux = y$ for $x$.
$x_n = y_n/A_{nn}$  % Assign the $n$-th solution.
For $i = n - 1: -1: 1$  % Loop through row $n - 1$ to row 1.
\[ S = 0 \] % Initialize sum counter for each variable.

For \( j = i + 1: n \)
\[ S = S + A_{ij} \times x_j \]
End
\[ x_i = (y_i - S) / A_{ii} \]
End

**A1.3.3 Cholesky Decomposition**

Like LU decomposition, Cholesky decomposition is direct method that solves systems of equations by separating the coefficient matrix \( A \) into two triangular matrices as \( A = LL^T \). The resulting triangular systems of equations are then solved successively: i.e., solve \( Ly = b \) for \( y \), and then solve \( L^Tx = y \) for \( x \). Although similar, the Cholesky method differs from LU decomposition in two respects: first, \( A \) is decomposed into \( A = LL^T \), where \( L \in \mathbb{R}^{n \times n} \) is a lower triangular matrix, and second, the method is only applicable to symmetric positive definite (SPD) matrices. Such systems are discussed alongside the conjugate gradient method in §A1.4.2.

Compared to the other direct solution methods discussed thus far, Cholesky decomposition is more efficient albeit limited to SPD systems. Additionally, the lower triangular matrix \( L \) can be constructed explicitly as:

\[ L_{jj} = \left( A_{jj} - \sum_{k=1}^{j-1} (L_{jk})^2 \right)^{1/2} \]  \[ \text{[A8]} \]
\[ L_{ij} = \frac{1}{L_{jj}} (A_{ij} - \sum_{k=1}^{j-1} l_{ik} L_{jk}), i > j \]  \[ \text{[A9]} \]

The above computations are implemented from left-right to top-down, starting at the \((1, 1)\) entry. Equation [A8] first assigns \( L_{jj} \), and then Equation [A9] loops through the row entries immediately below diagonal \( j \). A script is shown below for constructing the Cholesky matrix \( L \), excluding the triangular solves (note that they are the same as those used with LU decomposition):

\% Compute the diagonal elements of \( L, L_{jj} \).

For \( j = 1: n \)
\[ S = 0 \] % Initialize sum counter for each variable.

For \( k = 1: j - 1 \) % Loop through column 1 to column \( j - 1 \).
\[ S = S + (L_{jk})^2 \]
% Compute the below-diagonal elements of \( L \), \( L_{ij}, i > j \).

\begin{verbatim}
For \( i = j + 1: n \)  \% Loop through row \( j + 1 \) to row \( n \).
    \( S = 0 \)  \% Initialize sum counter for each variable.
    \textbf{For} \( k = 1: j - 1 \)  \% Loop through column 1 to column \( j - 1 \).
        \( S = S + L_{ik} \cdot L_{jk} \)
\textbf{End}
\textbf{End}
L_{ij} = (A_{ij} - S) / L_{jj}
\end{verbatim}

\subsection*{A1.3.4 Pivoting}

The direct methods discussed in the preceding sections work well and are easy to implement. However, they have one key shortcoming: the diagonals of \( A \) must be non-zero, otherwise the solution methodology will fail. To prevent this, rows, columns, or rows and columns can be interchanged to avoid dividing by zero diagonals. In this sense, the rearranged diagonal elements are referred to as \textit{pivots}, and the reordering process is called \textit{pivoting}.

Pivoting is performed prior to or during the implementation of a numerical solution. In either case, the pivoting actions are stored such that the solution can be returned to its original order upon completion. There are two forms of pivoting: \textit{partial pivoting} and \textit{complete pivoting}. The former involves reordering rows \textit{or} columns, while the latter considers the complete rearrangement of rows \textit{and} columns. As an example, consider naïve Gaussian elimination with partial pivoting while computing the solution. When operating on row \( i \) while reducing based on the diagonal element in column \( j \), we search all row entries \( k \) in column \( j \) for rows \( k \geq j \); once the largest absolute value is found in row \( k' \), rows \( i \) and \( k' \) are interchanged, and then the solution procedure continues.

Pivoting operations use \textit{permutation matrices} to reorder the rows and columns of \( A \). A permutation matrix \( P \in \mathbb{R}^{n \times n} \) comprises an \( \mathbb{R}^{n \times n} \) identity matrix whose rows have been reordered according to the desired pivoting procedure. The sorted matrix \( A' \in \mathbb{R}^{n \times n} \) is then
Appendix A: Review of Numerical Methods

\[ A' = PAP^T \]  \hspace{1cm} \text{[A10]}

The actual application of a permutation matrix is often expensive, so most solution algorithms simply store the row and column interchanges within permutation vectors.

To illustrate permutation operations, consider \( A \in \mathbb{R}^{3 \times 3} \) shown below. The permutation matrix \( P \) is taken as an identity matrix whose second and third rows are interchanged, so pivoting \( A \) via Equation [A10] exchanges its second and fourth rows and columns as follows:

\[
A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}
\]

\[
A' = PAP^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 6 & 5 \\ 2 & 0 & 3 \end{bmatrix}
\]

Because \( P^{-1} = P^T \), \( A' \) can be reverted to its original form as \( A = PA'P^T \).

A1.4 Iterative Methods for Systems of Linear Equations

A great many practical applications involve sparse systems of equations that contain far more zero elements than non-zero elements. The application of direct methods to these systems is often inefficient, for their solution methodologies perform numerous unnecessary operations with zero elements. Additionally, factorization techniques (such as Cholesky decomposition) lead to greater fill-in by removing some of the zero elements within the solution procedure, thereby destroying the original sparsity. Because they predominantly comprise matrix-vector operations, iterative methods are better suited to solving sparse systems of equations.

Unlike direct methods for solving systems of linear equations, iterative methods yield an approximate solution. Beginning with an initial solution estimate \( x^{(0)} \in \mathbb{R}^n \), a more accurate estimate \( x^{(1)} \in \mathbb{R}^n \) is computed, and then \( x^{(2)} \in \mathbb{R}^n \) is computed by using \( x^{(1)} \) as the initial estimate. This continues until the residual error \( r^{(k+1)} = b - Ax^{(k+1)} \in \mathbb{R}^n \) at iteration \( k + 1 \) is sufficiently small and thus the approximate solution \( x^{(k+1)} \in \mathbb{R}^n \) sufficiently accurate. Iterative methods are well suited to solving sparse systems of equations (see §A1.6.2), for they predominantly comprise matrix-vector computations. They also benefit from the ability to efficiently use sparse memory allocation techniques, thereby avoiding storing and operating with zero elements. Fundamental solution methods and overviewed below.
A1.4.1 Stationary Schemes

Stationary schemes are one approach to iteratively solving a system of linear equations. They approximate a system of equations with one that is more easily solved, such as a diagonal or triangular system. This leads to a numerical recursion of the form \( x^{(k+1)} = Tx^{(k)} + c \), where \( T \in \mathbb{R}^{n \times n} \) is an iteration matrix and \( c \in \mathbb{R}^n \). Stationary methods are named so because \( T \) and \( c \) do not change between iterations.

First we consider the Jacobi method. Using the Jacobi method, the coefficient matrix of a system of equations is replaced with \( A = D - (D - A) \), where \( D \in \mathbb{R}^{n \times n} \) comprises the diagonal elements of \( A \), and the proxy equation to be solved is \( Dx = (D - A)x + b \). Because diagonal systems of equations are easily solved, the recursion is

\[
x^{(k+1)} = (I - D^{-1}A)x^{(k)} + D^{-1}b
\]

At each iteration, only the first term on the right hand side must be updated because only \( x^{(k)} \) changes between iterations. As a result, the Jacobi method can be implemented quickly, but it typically converges slowly.

Instead of solving a diagonal system of equations with the Jacobi method, a triangular system of equations is solved with the Gauss-Seidel method. Consider substituting \( A = L - (L - A) \) for the coefficient matrix, where \( L \in \mathbb{R}^{n \times n} \) comprises the lower triangular elements of \( A \), including the diagonal. Using the Gauss-Seidel method, \( x^{(k+1)} \) is obtained by solving the triangular system

\[
Lx^{(k+1)} = (L - A)x^{(k)} + b
\]

With Equation [A12], only \( (L - A)x^{(k)} \) needs to be updated at each iteration. Compared to the Jacobi method, the Gauss-Seidel method converges faster because it involves solving a larger portion of \( A \) at each iteration (the Jacobi updates to \( \{x^{(k)}\}_i \) at each iteration are only local, meaning they do not consider the other solutions determined at iteration \( k + 1 \)). A more mathematically precise definition relates to the spectral radius of the methods’ iteration matrices.

Recall from §A1.2 that the spectral radius \( \rho(A) \) of a matrix \( A \in \mathbb{R}^n \) is its largest absolute eigenvalue. For stationary iterative methods where the matrix \( T \in \mathbb{R}^{n \times n} \) is repeatedly applied to \( x^{(k)} \in \mathbb{R}^n \) to obtain \( y = Tx^{(k)} \in \mathbb{R}^n \), the components of \( x^{(k)} \) are scaled in the direction of the eigenvectors of \( T \) according to its eigenvalues; thus, in order for a stationary method to converge, the largest eigenvalue of \( T \) must be less than unity, i.e., \( \rho(T) < 1 \). Furthermore, smaller \( \rho(A) \)
imply faster convergence. With the Jacobi method, \( T_J = I - D^{-1}A \), and for the Gauss-Seidel method, \( T_{GS} = I - L^{-1}A \). Because \( \rho(T_{GS}) < \rho(T_J) \), the Gauss-Seidel method will always converge faster given the same initial estimate. The efficiency of the Gauss-Seidel method can be made even better by improving the spectral properties of the iteration matrix \( T_{GS} \).

One means of reducing the spectral radius of the Gauss-Seidel method is to combine it (or other methods) with successive overrelaxation (SOR). Essentially, weights are assigned to \( x^{(k)} \) and the iterate computed via the Gauss-Seidel method to determine \( x^{(k+1)} \) as

\[ x^{(k+1)} = \omega \bar{x} + (1 - \omega) x^{(k)} \]  

where \( \omega = \) a weighting parameter, and \( \bar{x} = \) the result from the Gauss-Seidel method. A weighting of \( \omega = 0.5 \) weighs the previous and computed iterates equally, while \( \omega = 1 \) yields the original Gauss-Seidel method. As long as \( 0 < \omega < 2 \), the numerical method will converge. Of course, the speed at which it converges strongly depends on \( \omega \) and the nature of the problem (i.e., its eigenpairs). There is no single optimal value of \( \omega \), but \( 1 < \omega < 2 \) generally works best.

For the Gauss-Seidel method with SOR, the coefficient matrix is split as

\[ A = \left( \frac{1}{\omega} D + L' \right) - \left( \left( \frac{1}{\omega} - 1 \right) D - U' \right) \]  

where \( L' \) and \( U' \in \mathbb{R}^{n \times n} \) strictly comprise the lower and upper triangular elements of \( A \), exclusive of the diagonal elements. The iteration matrix is thus given as

\[ T_{SOR} = I - \left( \frac{1}{\omega} D + L' \right)^{-1} A \]  

Clearly, if \( \omega = 1 \), we get the iteration matrix for the normal Gauss-Seidel method. Comparatively, if \( \omega > 1 \), then the problem is overrelaxed, and if \( \omega < 1 \), it is underrelaxed.

The numerical methods discussed thus far (both direct and iterative) are simple, easy to implement, and well-studied in the literature. However, there are more powerful iterative methods, meaning more efficient yet also more complex. The preceding discussions still have purpose though, for direct and stationary methods are often used to construct preconditioners for more complex solvers, which often include the use of direct methods for simpler systems of equations. The following section continues the discussion of iterative methods by reviewing one particular approach for SPD systems of equations, the conjugate gradient method.
A1.4.2 SPD Matrices and the Conjugate Gradient Method

Some systems of equations feature distinct properties that make their solution easier. One such property is definiteness: a symmetric matrix $A \in \mathbb{R}^{n \times n}$ is definite if the scalar $|x^T A x| > 0$ for all non-zero $x \in \mathbb{R}^n$. Graphically, this means that $f(x) = \frac{1}{2} x^T A x - b^T x + c$ for $Ax = b$, a quadratic function, has a single extremum, and the extremum occurs at the solution $x$. From this, definite systems can be solved by considering $f(x)$ and $\partial f / \partial x$. One group of methods that encompass this philosophy are Krylov subspace methods. Such methods iteratively solve systems of linear equations by computing new solution iterates within vector spaces, called Krylov subspaces, not explored by previous iterates and residuals. The most prominent Krylov subspace method is the conjugate gradient (CG) method. It is only applicable to SPD systems (where $x^T A x > 0$ for all non-zero $x \in \mathbb{R}^n$), but it is powerful and widely used.

A comprehensive treatment of the CG method is beyond the scope here. Instead, only its implementation is reviewed. For further reading, Shewchuk (1994) provides a detailed explanation of the CG method, including illustrations, derivations, and convergence properties; Benzi et al. (2005) also discuss Krylov subspace methods (including CG) within the context of saddle point problems, a form of block-structured system of equations discussed later in §A1.6.1.

Given an SPD system of equations, the CG method minimizes $f(x)$ by using conjugated gradients to advance the solution iterates. Two vectors $x_1$ and $x_2 \in \mathbb{R}^n$ are $A$-orthogonal or conjugate on the basis of a matrix $A \in \mathbb{R}^{n \times n}$ if $x_1^T A x_2 = 0$; the CG method uses conjugated search directions to avoid searching for the solution in the same direction multiple times. An algorithm for implementing the CG method is shown below:

$$
\begin{align*}
    r^{(0)} &= b - Ax^{(0)} & \text{% Initialize the residual.} \\
    d^{(0)} &= r^{(0)} & \text{% Initialize search direction.} \\
    \text{For } k = 1: n \\
    \alpha^{(k)} &= \frac{(r^{(k)})^T r^{(k)}}{(d^{(k)})^T A d^{(k)}} & \text{% Compute new iterate step size.} \\
    x^{(k+1)} &= x^{(k)} + \alpha^{(k)} d^{(k)} & \text{% Compute new iterate.} \\
    r^{(k+1)} &= r^{(k)} - \alpha^{(k)} A d^{(k)} & \text{% Update residual vector.} \\
    \beta^{(k+1)} &= \frac{(r^{(k+1)})^T r^{(k+1)}}{(r^{(k)})^T r^{(k)}} & \text{% Compute new search direction step size.}
\end{align*}
$$
Appendix A: Review of Numerical Methods

\[ \mathbf{d}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta^{(k+1)} \mathbf{d}^{(k)} \] % Compute new search direction.
End

where \( \mathbf{r}^{(k)} \in \mathbb{R}^n \) is the residual at iteration \( k \); \( \mathbf{d}^{(k)} \in \mathbb{R}^n \) is the search direction at residual \( k \); \( \alpha^{(k)} \in \mathbb{R} \) is the solution iterate step size in search direction \( \mathbf{d}^{(k)} \) at iteration \( k \); and \( \beta^{(k+1)} \in \mathbb{R} \) is the search direction step size at iteration \( k \). The search direction vectors are determined such that all \( \mathbf{d}^{(k)} \) are conjugate on the basis of \( \mathbf{A} \).

After \( n \) iterations, the CG method will have explored the \( n \)-dimensional space of the problem. Accordingly, it will converge in \( n \) steps, but the algorithm can be terminated in fewer than \( n \) steps when the \( i \)-th solution iterate is sufficiently accurate. This requires a means of gauging the error of the approximate solution. Because the exact solution is unknown, the error on \( \mathbf{x}^{(k)} \) too is unknown, so instead the norm of the residual \( \| \mathbf{r}^{(k)} \| \) is used. A typical convergence criterion is

\[ \| \mathbf{r}^{(k)} \| < \varepsilon_{CG} \| \mathbf{b} \| \] [A16]

where \( \| \mathbf{r}^{(k)} \| \) is the vector norm of \( \mathbf{r}^{(k)} \), and \( \varepsilon_{CG} < 1 \) is an error tolerance. Note that the norm of some \( \mathbf{x} \in \mathbb{R}^n \) is given as \( \| \mathbf{x} \| = (\mathbf{x}^T \mathbf{x})^{1/2} \). The error tolerance should be sufficiently small (e.g., \( 10^{-5} \)), but not so small that it is close to machine precision.

The CG method is particularly well suited to solving large sparse systems of equations. It is efficient because it predominantly comprises matrix-vector operations, and the iterations can be terminated prematurely once the residual error is sufficiently small. Despite these advantages, Krylov subspace methods such as the CG method feature poor convergence when applied to systems with a wide-ranging spectrum. One means of improving this is to cluster the eigenvalues of the system of equations via preconditioning.

### A1.4.3 Preconditioned Conjugate Gradient Method

With stationary iterative methods, it was discussed how their convergence can be accelerated by improving the spectral properties of the iteration matrix \( \mathbf{T} \). Similar improvements can be realized with the CG method (and others) via preconditioning, where the resulting method is called the preconditioned CG (PCG) method. Preconditioning involves applying a preconditioner \( \mathbf{M} \in \mathbb{R}^{n \times n} \) to the system of equations \( \mathbf{A} \mathbf{x} = \mathbf{b} \) to cluster its eigenvalues. The transformed system \( \mathbf{M}^{-1} \mathbf{A} \mathbf{x} = \mathbf{M}^{-1} \mathbf{b} \) is easier to solve, for the range of \( \sigma(\mathbf{M}^{-1} \mathbf{A}) \) is smaller than \( \sigma(\mathbf{A}) \).

Like \( \mathbf{A} \), the preconditioner \( \mathbf{M} \) must be SPD. Despite this, the transformed coefficient matrix
Appendix A: Review of Numerical Methods

\( M^{-1}A \) is not necessarily SPD, even though \( A \) and \( M^{-1} \) are (note that because \( M \) is SPD, so too is \( M^{-1} \)). The CG method thus appears to be inapplicable; fortunately, there is a means of circumventing the matter, and it relies on the eigenpairs of \( M^{-1}A \). Consider an eigenvector \( v \in \mathbb{R}^n \) of \( M^{-1}A \) with eigenvalue \( \lambda \) where

\[
M^{-1}Av = \lambda v \tag{A17}
\]

For any SPD matrix \( M \), there exists a matrix \( E \in \mathbb{R}^{n \times n} \) such that \( M = EE^T \). Multiplying both sides of Equation [A17] by \( E^T \) and expanding \( M^{-1} = E^{-T}E^{-1} \) gives

\[
E^T(E^{-T}E^{-1})Av = \lambda E^T v \tag{A18}
\]

Next, the leading \( E^T \) and \( E^{-T} \) can be cancelled, and substituting \( A = AE^{-T}E^T \) gives

\[
(E^{-1}AE^{-T})(E^T v) = \lambda E^T v \tag{A19}
\]

This means that \( M^{-1}A \) and \( E^{-1}AE^{-T} \) have the same eigenvalues. Accordingly, the system \( Ax = b \) can be solved indirectly by solving \( (E^{-1}AE^{-T})\hat{x} = E^{-1}b \) via the CG method, where \( \hat{x} = E^T x \) and the coefficient matrix \( E^{-1}AE^{-T} \) is SPD.

An algorithm for implementing the preconditioned CG is shown below:

\[
\begin{align*}
    r^{(0)} &= b - Ax^{(0)} \quad \% \text{Initialize the residual.} \\
    d^{(0)} &= M^{-1}r^{(0)} \quad \% \text{Initialize search direction using the residual.} \\
    \text{For } i = 1:n \\
    \alpha^{(i)} &= \frac{(r^{(i)})^TM^{-1}r^{(i)}}{(d^{(i)})^TAd^{(i)}} \quad \% \text{Compute new iterate step size.} \\
    x^{(i+1)} &= x^{(i)} + \alpha^{(i)}d^{(i)} \quad \% \text{Compute new iterate.} \\
    r^{(i+1)} &= r^{(i)} - \alpha^{(i)}Ad^{(i)} \quad \% \text{Update residual vector.} \\
    \beta^{(i+1)} &= \frac{(r^{(i+1)})^TM^{-1}r^{(i+1)}}{(r^{(i)})^TM^{-1}r^{(i)}} \quad \% \text{Compute new search direction step size.} \\
    d^{(i+1)} &= M^{-1}r^{(i+1)} + \beta^{(i+1)}d^{(i)} \quad \% \text{Compute new search direction.}
\end{align*}
\]

End

Although the subject here is the preconditioned CG method, what constitutes a suitable preconditioner has yet to actually be discussed. The ideal preconditioner is \( M = A \), which
normalizes the eigenvalues of $A$; however, computing $M^{-1}$ is equivalent to directly solving $Ax = b$. Thus, we require preconditioners that approximate $A$ yet are easier to solve are. The simplest preconditioner is one comprising the main diagonal elements of $A$, a method often referred to as diagonal preconditioning or Jacobi preconditioning. Other more complicated approaches include incomplete LU decomposition and incomplete Cholesky factorization. The term “incomplete” refers to the fact that the preconditioners allow zero fill-in: that is, if $A_{ij}$ is zero, then, for example, the Cholesky preconditioner uses $L_{ij} = 0$ to preserve sparsity.

**A1.5 Non-Linear Systems of Equations and the Newton Method**

Until now, systems of linear equations were concerned. Of course, not every problem is linear, but understanding their solution is fundamental to solving systems of non-linear equations, which are more complex. Recall the a system of equations

$$f(x) = Ax - b = 0 \quad [A20]$$

from §A1.1, where $f(x): \mathbb{R}^n \to \mathbb{R}^n$. Equation [A20] is linear if and only if $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ are constant and the exponent on $x \in \mathbb{R}^n$ is unity. Otherwise, Equation [A20] is non-linear. Few (if any) systems of non-linear equations can be solved directly, so the only option is to rely on iterative non-linear solution methods that linearize Equation [A20].

Perhaps the most well-known approach for solving systems of non-linear equations is the Newton method (or the Newton-Raphson method). To derive the Newton method, consider the first order Taylor series expansion of Equation [A20] about an initial solution estimate $x^{(0)} \in \mathbb{R}^n$:

$$f(x^{(1)}) = f(x^{(0)}) + J^{(0)}(x^{(1)} - x^{(0)}) + R(O^2) \quad [A21]$$

where $f(x^{(0)})$ and $f(x^{(1)}): \mathbb{R}^n \to \mathbb{R}^n$, $x^{(1)} \in \mathbb{R}^n$ is the refined solution iterate, $J^{(0)} \in \mathbb{R}^{n \times n}$ is the Jacobian of $f(x^{(0)})$, and $R(O^2) \in \mathbb{R}^n$ is a second order remainder term. The $k$-th Newton step Jacobian matrix $J^{(k)}$ represents the gradient of $f(x^{(k)})$, and it has elements of $J_{ij}^{(k)} = \partial f_i^{(k)}/\partial x_j$.

For Equation [A20], $x$ is sought such that $f(x) = Ax - b = 0 \in \mathbb{R}^n$. Accordingly, Equation [A21] can be simplified by dropping $R(O^2)$ and setting $f(x^{(1)}) = 0$, thereby yielding

$$J^{(0)}s^{(1)} = -f(x^{(0)}), \quad s^{(1)} = x^{(1)} - x^{(0)} \quad [A22]$$

where $s^{(1)} \in \mathbb{R}^n$ is the step size at Newton step 1. The step size is determined by solving the linear system Equation [A22] using any of the methods discussed in the preceding sections. This
exercise is repeated to determine $s^{(2)}$ and $x^{(2)}$, $s^{(3)}$ and $x^{(3)}$, and so forth. The recursive formulation of the Newton method is thus

$$f^{(k)}s^{(k+1)} = -f(x^{(k)}), \quad s^{(k+1)} = x^{(k+1)} - x^{(k)}$$ \[A23\]

Equation [A23] is solved for successive Newton steps until the solution converges within an acceptable level of error. Similar to the CG method, the error on $x^{(k)}$ cannot be determined without knowing the exact solution, but the Newton step residual error $r_N^{(k)} = f(x^{(k)}) \in \mathbb{R}^n$ can be used to gauge the accuracy of the approximate solution. A similar convergence criterion for the approximate solution to Equation [A20] can be adopted, i.e.,

$$\|r_N^{(k)}\| < \varepsilon_N \|b\|$$ \[A24\]

where $\varepsilon_N < 1$ = an error tolerance that is greater than machine precision.

Considering Equation [A23], $s^{(k+1)}$ can be determined using direct or iterative methods. Using the former, the algorithm is called an exact Newton method, for $s^{(k+1)}$ is determined exactly (to the limits of floating point arithmetic). Comparatively, if an iterative method such as the PCG method is applied, the linear solve iterations can be terminated prematurely once they are sufficiently accurate. Such an approach is called an inexact Newton method (Dembo et al., 1982), the purpose of which is to facilitate an efficient solution.

In using an inexact Newton algorithm, Equation [A23] is solved iteratively for $s^{(k+1)}$ until

$$\|r_{iter}^{(k+1,l)}\| < \eta^{(k+1)} \|f(x^{(k+1,l)})\|$$ \[A25\]

where $r_{iter}^{(k+1,l)} = f^{(k+1,l)}s^{(k+1,l)} + f(x^{(k+1,l)}) \in \mathbb{R}^n$ is the residual at iteration $l$ of Newton step $k + 1$, and $\eta^{(k+1)}$ = a forcing sequence that may vary between the Newton steps. Setting $\eta^{(k+1)} = 0$ results in an exact solution to the linear system Equation [A23], i.e., the exact Newton method. A suitable approach is to allow $\eta^{(k+1)}$ to vary according to the Newton step as

$$\eta^{(k+1)} = \min \left( \eta, \eta \|r_N^{(k)}\| \right)$$ \[A26\]

where $\eta$ = the iteration convergence tolerance (typically small yet larger than $\varepsilon_N$). For the early Newton steps where $\|r_N^{(k)}\|$ is large, $\eta^{(k+1)}$ takes the value of $\eta$. At later iterations when the approximate solution to Equation [A20] nears convergence, the iteration tolerance becomes stricter such that the Newton step $s^{(k+1)}$ is computed exactly.
While powerful, the Newton method features some limitations. In order for the algorithm to converge, the initial solution estimate should not be far from the true solution: if there are local extrema between the true solution and the initial estimate, convergence is not guaranteed. Furthermore, the Jacobian of Equation [A20] must be computable. If it cannot be computed (or if it is too expensive to compute), other approaches such as the secant method can be employed.

A1.6 Other Topics

This section has thus far concerned the numerical solution of systems of equations. Additionally, there are numerous other facets worth studying, but only two more are discussed below. The first, saddle point problems, concern structured systems of linear equations. Saddle point problems commonly arise in fluid problems, including the mathematical models formulated in this thesis. Understanding their properties aids their efficient solution. Many applications, including saddle point problems, comprise large, sparse systems of equations, so it is also worth reviewing techniques for sparse memory allocation and sparse matrix operations.

A1.6.1 Saddle Point Problems

Saddle point problems involve systems of linear equations having a specific $2 \times 2$ block structure where the coefficient matrix $A \in \mathbb{R}^{n \times n}$ consists of four block sub-matrices. The key characteristics of saddle point problems (as they pertain to this thesis) are briefly overviewed here, for the mathematical formulations of this thesis are saddle point problems. The material is referenced from Benzi et al. (2005), who provide a comprehensive treatment of the topic.

Saddle point problems feature the $2 \times 2$ block structure shown below:

$$\begin{bmatrix} A & B_1^T \\ B_2 & -C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad Au = b \tag{A27}$$

where $A \in \mathbb{R}^{m+n \times m+n}$ is presumably non-singular; $A \in \mathbb{R}^{m \times m}$, $B_1$ and $B_2 \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{n \times n} = \text{coefficient matrices}$; $u \in \mathbb{R}^{m+n}$, $x \in \mathbb{R}^m$, and $y \in \mathbb{R}^n = \text{solution vectors}$; $b \in \mathbb{R}^{m+n}$, $b_1 \in \mathbb{R}^m$, and $b_2 \in \mathbb{R}^n = \text{constant vectors}$; and $m \geq n$. The coefficient matrix $C$ may be a zero matrix, but $A$, $B_1$, and $B_2$ must be non-zero.

Equation [A27] can be solved in two ways. The whole system can be solved simultaneously (a “coupled” approach), or individual expressions for $x$ and $y$ can be obtained by expanding and manipulating the row equations (a “segregated” approach). The latter yields
\[ x = \mathcal{A}^{-1}(b_1 - B_1^Ty) \]  \hspace{1cm} \text{[A28]}

and

\[ y = S^{-1}(b_2 - B_2\mathcal{A}^{-1}b_1) \]  \hspace{1cm} \text{[A29]}

where \( S = -(B_2\mathcal{A}^{-1}B_1^T + C) \in \mathbb{R}^{n \times n} \) is the Schur complement. With the above, the problem appears to be readily addressed. However, the solution for \( x \) from Equation [A28] requires \( y \), and \( y \) must be obtained by solving the \( \mathbb{R}^{n \times n} \) system of equations \( S y = b_2 - B_2\mathcal{A}^{-1}b_1 \). The solutions are no so easily obtained. Nonetheless, the solutions can indeed be determined, and they can be determined efficiently by understanding the properties of the particular problem.

Consider the case where \( \mathcal{A} \) is SPD, \( B = B_1 = B_2 \), and \( C \neq 0 \) is symmetric positive semi-definite (i.e., \( x^TCx \geq 0 \) for any \( x \in \mathbb{R}^n \)). Such conditions are common in incompressible fluid flow problems. The negative Schur complement \( -S \) is thus symmetric positive semi-definite, and it is SPD (and thus non-singular) if and only if \( \ker(C) \cap \ker(B^T) = 0 \). For \( A \) to be non-singular, either \( B \) must have full rank, or \( C \) must be SPD. Because the latter precludes the possibility of \( C = 0 \), it will be assumed that \( B \) indeed has full rank. If the negative Schur complement is SPD, reliable numerical methods such as Cholesky factorization or the PCG method can be applied to solve Equations [A28] and [A29].

Owing to their structure, saddle point problems can be readily factorized. Continuing from the above, the lower-diagonal-upper decomposition of Equation [A20] is

\[ A = \begin{bmatrix} I & 0 \\ B\mathcal{A}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{A} & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} \mathcal{A}^{-1}B^T \\ I \end{bmatrix} \]  \hspace{1cm} \text{[A30]}

The equivalent LU factorization is given as

\[ A = LU = \begin{bmatrix} I & 0 \\ B\mathcal{A}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{A} & \mathcal{A}^{-1}B^T \\ 0 & S \end{bmatrix} \]  \hspace{1cm} \text{[A31]}

Further to the above, the inverse of \( A \) is explicitly given as

\[ A^{-1} = \begin{bmatrix} \mathcal{A} & B_1^T \\ B_2 & -C \end{bmatrix}^{-1} = \begin{bmatrix} \mathcal{A}^{-1} + \mathcal{A}^{-1}B_1^TS^{-1}B\mathcal{A}^{-1} & -\mathcal{A}^{-1}B_1^TS^{-1} \\ -S^{-1}B\mathcal{A}^{-1} & S^{-1} \end{bmatrix} \]  \hspace{1cm} \text{[A32]}

Solving a system of equations with any of the above requires solving a system having \( S \) as the coefficient matrix, which is analogous to solving Equation [A29]. Nonetheless, the factorizations above are helpful for constructing preconditioners.
A1.6.2 Sparse Systems of Equations

Sparse systems of equations were briefly mentioned at the introduction to iterative methods in §A1.4. These types of equations feature many more zero elements than non-zero elements in the coefficient matrix, and they commonly arise in practical applications. A key factor affecting the solution efficiency of sparse systems is the way in which they are handled in computer memory. A few sparse matrix storage formats are reviewed here. Indeed, other factors affect computational efficiency (e.g., array access patterns), but such material is better left to standard texts (e.g., George and Liu, 1986; Davis, 2006).

Consider the $\mathbb{R}^{m \times n}$ symmetric matrix

$$
A = \begin{bmatrix}
1 & 0 & 1 & 0 & 0 \\
0 & 2 & 0 & 1 & -1 \\
1 & 0 & 3 & 0 & 0 \\
0 & 1 & 0 & 4 & 0 \\
0 & -1 & 0 & 0 & 5
\end{bmatrix}
$$

which has 25 elements. Of those, only $n_{NZ} = 11$ are non-zero elements, meaning $A$ has a sparsity of $n_{NZ}/m \times n = 0.44$. If $A$ were stored in its dense format as shown above, all 25 elements (both zero and non-zero) would be written to memory, and any matrix-vector operation with $A$ and some $x \in \mathbb{R}^5$ would have an operation count of roughly $2m \times n$. Given the number of zero elements, $n_{NZ}$, this representation is clearly inefficient.

As an alternative, $A$ can be stored using \textit{sparse memory allocation}. Sparse memory allocation techniques only store and operate with the non-zero elements of a sparse matrix; the remaining entries are disregarded because they are, by definition, all zeros. In this way, computer memory is freed up for other purposes, thereby allowing faster matrix operations for large systems. One such option is to store $A$ using the \textit{coordinate list (COO) format}. The COO storage format stores the row indices, the column indices, and the values of the non-zero elements. The elements can be stored in \textit{row major order}, where they are ordered by row, or the values can be stored in \textit{column major order}, where they are ordered by columns. Because matrix-vector operations will be performed, the former will be used. For $A$, the row major COO storage is

$$
r = [1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 3 \quad 3 \quad 4 \quad 4 \quad 5 \quad 5]^T
$$

$$
c = [1 \quad 3 \quad 2 \quad 4 \quad 5 \quad 1 \quad 3 \quad 2 \quad 4 \quad 2 \quad 5]^T
$$

$$
v = [1 \quad 1 \quad 2 \quad 1 \quad -1 \quad 1 \quad 3 \quad 1 \quad 4 \quad -1 \quad 5]^T
$$

where $r \in \mathbb{R}^{n_{NZ}}$ contains the row indices of the non-zero elements, $c \in \mathbb{R}^{n_{NZ}}$ contains the
column indices of the non-zero elements, and \( \mathbf{v} \in \mathbb{R}^{\text{NNZ}} \) contains the non-zero element values.

With the COO format, the matrix-vector operation count is approximately \( 2n_{\text{NNZ}} \), a much smaller amount than that for dense matrix storage. A simple algorithm for computing the matrix-vector product is also shown below:

\[
\% \text{ Compute } \mathbf{y} = \mathbf{A} \mathbf{x}, \text{ where } \mathbf{A} \text{ is stored in row major COO format.}
\]

\[
\text{For } l = 1:n_{\text{NNZ}} \\
\quad i = r_l \quad \% \text{ Row index of non-zero element } l \text{ of } \mathbf{A}. \\
\quad j = c_l \quad \% \text{ Column index of non-zero element } l \text{ of } \mathbf{A}. \\
\quad y_i = y_i + v_l \cdot x_j \quad \% \text{ Update the solution for } y_i.
\]

\text{End}

From the above, the matrix-vector operation count is the same regardless of whether \( \mathbf{A} \) is stored in row or column major order. If column major order were used, the matrix-vector computation would still loop through the non-zero elements of \( \mathbf{A} \), but they would be in a different order. Despite this, row or column major order is indeed important when a particular row or column is needed. Consider storing \( \mathbf{A} \) in row major COO format as shown above. If a particular row is needed, the corresponding values can be pulled easily, for the row indices in \( \mathbf{r} \) are stored in ascending order (that is, simply search for the first and last instances of a row index of 3 and pull the corresponding values). Now, say the third column were needed; a search algorithm must loop through all of the elements of the unordered \( \mathbf{c} \) to find all \( i \) with \( c_i = 3 \). This is less efficient.

Further reductions in the memory required to store an \( \mathbb{R}^{m \times n} \) matrix are still possible. Rather than store the row and column indices for each non-zero element, \textit{compressed sparse row} (CSR) storage indexes the number of non-zero elements within each row in. Its parallel, \textit{compressed sparse column} (CSC), uses column major ordering. Compared to COO, the compressed formats are better suited to matrix-vector operations, particularly those involving specific rows or columns, but they are less efficient when non-zero values are added or removed.

The CSR storage of an \( \mathbb{R}^{m \times n} \) matrix uses \( \mathbf{c} \) and \( \mathbf{v} \) along with a third vector, \( x_{\text{CSR}} \in \mathbb{R}^{m+1} \).

This term contains the cumulative number of non-zero values over the rows, and its elements are

\[
x_{\text{CSR}}_i = \begin{cases} 
0 & i = 1 \\
 x_{\text{CSR}}_{i-1} + n_{R_i} & i > 1
\end{cases}
\]  \hspace{1cm} \text{[A33]}

202
where \( n_{R_i} \) is the number of non-zero elements in row \( i \). Conveniently, the last element of \( x_{CSR} \) is the number of non-zero elements, \( n_{NNZ} \). Using CSR storage, the non-zero elements in the \( i \)-th row of an \( \mathbb{R}^{m \times n} \) matrix are \( v_{j,k} \) with column indices \( c_{j,k} \), where \( j = x_{CSR_i} + 1 \) and \( k = x_{CSR_{i+1}} \).

The CSR storage of \( A \) is

\[
\begin{align*}
x_{CSR} &= [0 \ 2 \ 5 \ 7 \ 9 \ 11]^T \\
c &= [1 \ 3 \ 2 \ 4 \ 5 \ 1 \ 3 \ 2 \ 4 \ 2 \ 5]^T \\
v &= [1 \ 1 \ 2 \ 1 \ -1 \ 1 \ 3 \ 1 \ 4 \ -1 \ 5]^T
\end{align*}
\]

Compared to COO storage, the CSR representation is even smaller, hence the name \textit{compressed}. The pseudocode below illustrates how the matrix-vector product \( y = Ax \) is computed when the coefficient matrix \( A \) is stored in CSR format:

\[
\begin{align*}
&\text{R = rows}(A) \quad \% \text{Number of rows in } A. \\
&\text{for } i = 1:R \quad \% \text{Loop through the rows of } A. \\
&\quad j = x_{CSR_i} + 1 \quad \% \text{Start index of non-zero elements in row } i \\
&\quad k = x_{CSR_{i+1}} \quad \% \text{End index of non-zero elements in row } i \\
&\quad \text{for } l = j: k \\
&\quad \quad y_l = y_l + v_l \cdot x_{c_l} \\
&\quad \text{end} \\
&\text{end}
\end{align*}
\]

The CSC storage format is similar to CSR. It uses \( r \) and \( v \) along with \( x_{CSC} \in \mathbb{R}^{n+1} \), a container for the cumulative number of non-zero elements over the columns. Its elements are

\[
x_{CSC_i} = \begin{cases} 
0 & i = 1 \\
x_{CSC_{i-1}} + n_{c_i} & i > 1 
\end{cases} \tag{A34}
\]

where \( n_{c_i} \) is the number of non-zero elements in column \( i \). Like CSR, the last element of \( x_{CSC} \) is \( n_{NNZ} \). Using CSC storage, the non-zero elements in the \( j \)-th column are \( v_{i,k} \) with row indices \( r_{i,k} \), where \( i = x_{CSC_j} + 1 \) and \( k = x_{CSC_{j+1}} \). The CSC storage of \( A \) is
Appendix A: Review of Numerical Methods

\[ \mathbf{x}_{\text{CSC}} = \begin{bmatrix} 0 & 2 & 5 & 7 & 9 & 11 \end{bmatrix}^T \]
\[ \mathbf{r} = \begin{bmatrix} 1 & 3 & 2 & 4 & 5 & 1 & 3 & 2 & 4 & 2 & 5 \end{bmatrix}^T \]
\[ \mathbf{v} = \begin{bmatrix} 1 & 1 & 2 & 1 & -1 & 1 & 3 & 1 & 4 & -1 & 5 \end{bmatrix}^T \]

The CSR and CSC row and column terms \( \mathbf{x}_{\text{CSR}} \) and \( \mathbf{x}_{\text{CSC}} \) are in fact the same because \( \mathbf{A} \) is symmetric. For the same reason, the values stored in \( \mathbf{v} \) are also the same for both storage formats.

\[ \% \text{Compute } \mathbf{y} = \mathbf{A} \mathbf{x}, \text{ where } \mathbf{A} \text{ is stored in CSC format.} \]
\[ \mathbf{C} = \text{columns}(\mathbf{A}) \]
\[ \% \text{Number of columns in } \mathbf{A}. \]
\[ \text{For } i = 1: \mathbf{C} \]
\[ \% \text{Loop through the columns of } \mathbf{A}. \]
\[ i = x_{\text{CSR}} j + 1 \]
\[ \% \text{Start index of non-zero elements in row } j \]
\[ k = x_{\text{CSR}} j + 1 \]
\[ \% \text{End index of non-zero elements in row } j \]
\[ \text{For } l = j: k \]
\[ y_{r_j} = y_{r_j} + v_l \cdot x_i \]
\[ \text{End} \]
\[ \text{End} \]

The selection of which storage format to use depends on the application. If non-zero elements are to be frequently added or removed from a sparse matrix, then the COO format is the most suitable. Comparatively, CSR and CSC are more effective when individual rows and columns must be pulled from a matrix; the former is efficient for matrix-matrix operations, which can be treated as a sequence of matrix-vector operations. Beyond the discussions here, further reductions in memory allocation can be achieved by tailoring the storage format to structure of a matrix.

A1.7 Summary

Systems of equations arise in various shapes and forms. Their properties vary largely, from size to sparsity and ordering to definiteness. In solving systems of equations, it is often more efficient to treat the problem as \( \mathbf{A} \mathbf{x} = \mathbf{b} \) rather than compute \( \mathbf{A}^{-1} \). Direct and iterative techniques for solving equations of this form were reviewed. The former yield exact solutions (exact in the absence of rounding error), while the latter provide an approximate (albeit reasonably accurate) solution. With techniques such as the PCG method, preconditioning begins to blur the boundary between direct and iterative solutions. Using concepts from the aforementioned, the solution of
systems of non-linear equations via the Newton method were explored.

Ultimately, the efficacy of one numerical method or storage scheme over another depends on the problem at hand. Accordingly, it is not enough to simply understand the application of a numerical method; the reader should also be familiar with the numerical characteristics of the problem itself. The following sections build upon the material covered thus far by exploring the solution of systems of differential equations.

A2. Numerical Solution of Systems of ODE IVPs

*Systems of linear differential equations* represent systems of equations that, in addition to having an unknown $x \in \mathbb{R}^n$, include a differential of $x$. Solution methods for such systems must simultaneously consider both $x$ and its differential. *Analytical solutions* to ODEs are obtained through the application of *analytical methods*, which involve direct integration. Although they yield exact solutions (exact in a numerical sense), the application of analytical methods is limited to select problems; numerical techniques are thus needed to solve those expressions that cannot be solved analytically. Unlike analytical solutions, *numerical solutions* (obtained via the application of a numerical method) provide an inexact solution. That is, they represent an approximation of the true solution.

The subject here is the solution of systems of linear ODEs via numerical methods. As we saw in §A1.4 with iterative methods for systems of linear equations, approximate methods feature varying accuracy, stability, and convergence properties, so it is of interest to study and understand them. The material of this section is based on Strang (1986b) and Epperson (2002), both of whom provide detailed discussions with helpful examples. We begin by defining the problem of interest, IVPs, and then common numerical methods for solving these problems are reviewed, including their accuracy and stability characteristics.

A2.1 Initial Value Problems

Problems described via ODEs are classified according to their prescribed conditions and linearity. For *boundary value problems* (BVPs), a solution is sought subject to conditions at the boundaries of the domain: these are referred to as *boundary conditions*. *Initial value problems* (IVPs) also require a solution over the domain, but instead of boundary conditions, *initial conditions* are given. These represent the starting point of the solution to an IVP. A problem may also be linear or non-linear, the latter of which is characterized by a non-linear system of ODEs.
Appendix A: Review of Numerical Methods

For IVPs, the independent variable is typically time \( t \) (used herein). A multi-dimensional IVP can be represented as

\[
\frac{d}{dt}(y) = f(t, y) = A \cdot y + b
\]  

where \( y(t) \in \mathbb{R}^n = \) the solution vector; \( f(t, y): \mathbb{R}^n \rightarrow \mathbb{R}^n = \) the time rate of change of \( y(t) \); \( A \in \mathbb{R}^{n \times n} = \) a coefficient matrix; and \( b(t) \in \mathbb{R}^n = \) a forcing vector. If \( A = A(t) \), the problem is linear, but if \( A = A(t, y) \), then Equation [A35] is non-linear.

To obtain the numerical approximation to \( y \), a time marching approach is used. Starting with the initial condition \( y^{(0)} = y(t = 0) \), the numerical solution \( y^{(1)} \) at time \( t_1 = t_0 + \Delta t \) is computed using a numerical method, where \( \Delta t \) is the integration interval or time step (s). The numerical solution \( y^{(2)} \) at time \( t_2 = t_1 + \Delta t \) is then computed using \( y^{(1)} \) as the starting point and so on. More generally, the numerical solution \( y^{(n+1)} \) at time \( t_{n+1} = t_n + \Delta t \) is calculated using the solution \( y^{(n)} \) from the previous step. They two key factors here are the choice of \( \Delta t \) and the numerical method, and both directly affect the accuracy of the numerical solution (i.e., how well the numerical solution represents the true solution).

As an illustrative example of how the numerical method and \( \Delta t \) affect the solution, consider the scalar IVP \( dy/dt = -10y + 10 \) with the initial condition \( y(0) = 0 \). This represents an exponential decay problem where \( y(t) \) begins growing rapidly, and then it eventually reaches a steady-state value of unity. The solutions from two numerical methods are plotted against the exact solution in Figure A1 for two different time steps.

Clearly the numerical method and \( \Delta t \) affect the quality of the solution. Both approaches used here have the same order of accuracy, yet they provide different solutions due to their underlying differences. The degree to which the quality of the solution improves by reducing \( \Delta t \) also depends on the particular numerical method. Because the specific approach and \( \Delta t \) are key to obtaining an numerically accurate and numerically stable solution, it is of interest to study different methods and their performance. It is also illustrated in §A2.3 that the stability properties of a numerical method are key to the solution of so-called stiff differential equations. Some of the more common numerical methods are reviewed first though.
Appendix A: Review of Numerical Methods

A2.2 Common Solution Schemes

Many numerical methods exist for solving ODE IVPs. Each features different accuracy and stability characteristics, so it is important to select both the method and $\Delta t$ with these in mind. Some of the more common numerical methods are briefly overviewed here; the reader is directed to Strang (1986b) and Epperson (2002) for more detailed treatments.

Before proceeding, it is worth reviewing a few definitions. The order of accuracy of a method, denoted by $O(\Delta t^m)$, represents the level of error associated with the method for an integration interval $\Delta t$, where $m$ is the degree of accuracy. For example, an $O(\Delta t^2)$ accurate numerical method has second order accuracy: this means that the numerical error is proportional to $\Delta t^2$. Some approaches determine $y^{(n+1)}$ solely based on $y^{(n)}$. These are called explicit methods, for they compute $y^{(n+1)}$ as an explicit function of $y^{(n)}$. In contrast, implicit methods compute $y^{(n+1)}$ based on both $y^{(n)}$ and $y^{(n+1)}$. Although generally less efficient, implicit methods feature greater numerical stability (they are less prone to a numerical solution divergence). In addition to being explicit or implicit, a numerical method may use information from a single integration step or multiple integration steps. The latter are termed multi-step methods, which typically have higher accuracy and better stability. Each of these concepts characterizes the performance and thus suitability of a particular numerical method, and they are explored below.

A2.2.1 Single Step Methods

The simplest single-step method is the Euler method. The Euler method is an $O(\Delta t)$ accurate
Appendix A: Review of Numerical Methods

(it has first order accuracy) explicit scheme that calculates $y^{(n+1)}$ as a function of $y^{(n)}$ and $f(t_n, y^{(n)})$. By using a first order finite difference approximation of the differential term, the recursive time marching solution for the Euler method is

$$y^{(n+1)} = y^{(n)} + \Delta t \cdot f(t_n, y^{(n)})$$  \[A36\]

Although simple, the Euler method is only first order accurate. As an improvement, predictor-corrector schemes can be used. These techniques first predict the solution, $\bar{y}$, and then they compute $y^{(n+1)}$ using a corrected estimate of $f(t, y)$ on the interval $[t, t + \Delta t]$. One example of such an approach is the $O(\Delta t^2)$ accurate modified Euler method, which uses the Euler method to calculate the predictor. Its recursion is

$$y^{(n+1)} = y^{(n)} + \frac{1}{2} \Delta t \cdot \left( f(t_n, y^{(n)}) + f(t_{n+1}, \bar{y}) \right)$$  \[A37\]

$$\bar{y} = y^{(n)} + \Delta t \cdot f(t_n, y^{(n)})$$

Another example of a predictor-corrector scheme is the second order Runge-Kutta (RK2) method. It generalizes the modified Euler method by assigning weightings to the predictor and corrector derivatives. The RK2 method recursion is

$$y^{(n+1)} = y^{(n)} + c_1 \Delta t \cdot f(t_n, y^{(n+1)}) + c_2 \Delta t \cdot f(t_n + \alpha \Delta t, \bar{y})$$  \[A38\]

$$\bar{y} = y^{(n)} + \beta \Delta t \cdot f(t_n, y^{(n)})$$

where $\alpha$, $\beta$, $c_1$, and $c_2$ = weighting parameters. Typically $c_1$ is chosen, and then $c_2$ is taken as $c_2 = 1 - c_1$ with $\alpha = \beta = 1/2c_2$. Like the modified Euler method, the RK2 scheme is explicit and $O(\Delta t^2)$ accurate.

An even more accurate single-step explicit scheme is the fourth order Runge-Kutta method (RK4). Like the RK2 method, weightings are assigned to the derivative terms, but the RK4 method uses a greater number of predictor-corrector computations. The RK4 method is $O(\Delta t^4)$ accurate, and its numerical computations are given by

$$y^{(n+1)} = y^{(n)} + \frac{1}{6} (K_1 + K_2 + K_3 + K_4)$$

$$K_1 = \Delta t \cdot f(t_n, y^{(n)})$$

$$K_2 = \Delta t \cdot f(t_{n+1/2}, y^{(n)} + \frac{1}{2} K_1)$$

$$K_3 = \Delta t \cdot f(t_{n+1/2}, y^{(n)} + \frac{1}{2} K_2)$$

$$K_4 = \Delta t \cdot f(t_{n+1}, y^{(n)} + K_3)$$  \[A39\]
where $K_1, K_2, K_3,$ and $K_4 \in \mathbb{R}^n$ = intermediate vectors. The RK4 method is relatively simple, accurate, and readily implemented, but it involves a greater number of computations.

The single-step numerical methods discussed thus far have relied on explicit numerical integration. That is, the solution $y^{(n+1)}$ is computed explicitly based on $y^{(n)}$ and $f(t_n, y^{(n)})$. Comparatively, implicit schemes solve for $y^{(n+1)}$ while simultaneously considering $y^{(n+1)}$ and $f(t_{n+1}, y^{(n+1)})$. The backward Euler method is one example of an implicit scheme. It is $O(\Delta t)$ accurate, and its numerical recursion is

$$y^{(n+1)} = y^{(n)} + \Delta t \cdot f(t_{n+1}, y^{(n+1)})$$  \[A40\]

Another common implicit method is the $O(\Delta t^2)$ accurate trapezoid rule. It represents a combination of the Euler method and backward Euler method by solving the implicit expression

$$y^{(n+1)} = y^{(n)} + \frac{1}{2} \Delta t \cdot \left( f(t_n, y^{(n)}) + f(t_{n+1}, y^{(n+1)}) \right)$$  \[A41\]

Equations [A40] and [A41] are not as easily solved as Equations [A36] through [A39]. The interdependent $y^{(n+1)}$ and $f(t_{n+1}, y^{(n+1)})$ terms must be resolved simultaneously, a typical feature of implicit schemes. In the case of linear ODEs, the problem can often be rearranged to obtain a direct expression for $y^{(n+1)}$, but this often requires solving a system of linear equations.

### A2.2.2 Multi-Step Methods

Compared to single-step approaches, multi-step numerical methods rely on the solutions from multiple previous time steps to compute $y^{(n+1)}$. To determine $y^{(n+1)}$, a multi-step method may use the solutions from times $t_n, t_{n-1},$ and $t_{n-2}$. Like single-step methods, multi-step methods feature varying degrees of numerical accuracy that depend on the particular method and $\Delta t$, but their accuracy is additionally affected by the number of steps. Multi-step methods are not concerned in this thesis, though familiarity helps provide context for single-step methods.

Two common multi-step methods are the Adams-Bashforth and Adams-Moulton methods (collectively referred to as the Adams methods). They are based on the exact integration of interpolating polynomials that approximate $f(t, y)$, and their numerical accuracy reflects the number of multi-steps considered (e.g., the method is $O(\Delta t^4)$ accurate if four steps are used). The numerical recursion for the Adams-Bashforth method is

$$y^{(n+1)} = y^{(n)} + \sum_{i=0}^{p} k_i \cdot f(t_{n-i}, y^{(n-i)})$$  \[A42\]
where \( p + 1 \) = the number of steps, \( i \) = the step index relative to the solution step \( (y^{(n+1)}) \) corresponds to \( i = -1 \), and \( k_i \) = a coefficient. The Adams-Bashforth method is an explicit method, and its coefficients \( k_i \) are tabulated in Table A1.

### Table A1: Adams-Bashforth method coefficients

<table>
<thead>
<tr>
<th>No. Steps</th>
<th>( p )</th>
<th>( k_0 )</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( k_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>( \Delta t )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>( \frac{3}{2} \Delta t )</td>
<td>( \frac{1}{2} \Delta t )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>( \frac{23}{12} \Delta t )</td>
<td>( -\frac{16}{12} \Delta t )</td>
<td>( \frac{5}{12} \Delta t )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>( \frac{55}{24} \Delta t )</td>
<td>( -\frac{59}{24} \Delta t )</td>
<td>( \frac{37}{24} \Delta t )</td>
<td>( -\frac{9}{24} \Delta t )</td>
</tr>
</tbody>
</table>

If the Adams-Bashforth steps are shifted to include that at time \( t_{n+1} \), one gets the Adams-Moulton method. This results in an implicit formulation, where \( y^{(n+1)} \) is determined by solving

\[
y^{(n+1)} = y^{(n)} + \sum_{i=-1}^{p-1} k_i \cdot f(t_{n-i}, y^{(n-i)}) \tag{A43}
\]

Equations [A42] and [A43] are similar, with the key difference being that the latter is implicit. The Adams-Moulton coefficients are summarized in Table A2.

### Table A2: Adams-Moulton method coefficients

<table>
<thead>
<tr>
<th>No. Steps</th>
<th>( p )</th>
<th>( k_{-1} )</th>
<th>( k_0 )</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>( \Delta t )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>( \frac{1}{2} \Delta t )</td>
<td>( \frac{1}{2} \Delta t )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>( \frac{5}{12} \Delta t )</td>
<td>( \frac{8}{12} \Delta t )</td>
<td>( -\frac{1}{12} \Delta t )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>( \frac{9}{24} \Delta t )</td>
<td>( \frac{19}{24} \Delta t )</td>
<td>( -\frac{5}{24} \Delta t )</td>
<td>( \frac{1}{24} \Delta t )</td>
</tr>
</tbody>
</table>

The last group of multi-step methods we will consider is the backward difference formula (BDF) family. Like the Adams methods, the BDF methods use interpolating polynomials, but they are applied to \( y(t) \) instead of \( f(t, y) \). Of the two families of methods, the BDF schemes are more recent. They were developed to resolve stiff systems of equations, which are explored more in §A2.3. Using a BDF method, \( y^{(n+1)} \) is computed by solving the expression

\[
y^{(n+1)} = \sum_{i=0}^{p-1} \mu_i \cdot y^{(n-i)} + \nu \cdot f(t_{n+1}, y_{n+1}) \tag{A44}
\]
where \( \mu_i \) and \( v = \) coefficients according to Table A3. Due to the presence of \( f(t_{n+1}, y^{(n+1)}) \), all BDF methods are implicit, and their accuracy depends on the number of steps.

**Table A3: Backward difference formula coefficients**

<table>
<thead>
<tr>
<th>No. Steps</th>
<th>( p )</th>
<th>( v )</th>
<th>( \mu_0 )</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>( \mu_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>( \Delta t )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>( \frac{2}{3} \Delta t )</td>
<td>( \frac{4}{3} )</td>
<td>( -\frac{1}{3} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>( \frac{6}{11} \Delta t )</td>
<td>( \frac{18}{11} )</td>
<td>( -\frac{9}{11} )</td>
<td>( \frac{2}{11} )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>( \frac{12}{25} \Delta t )</td>
<td>( \frac{48}{25} )</td>
<td>( -\frac{36}{25} )</td>
<td>( \frac{16}{25} )</td>
<td>( -\frac{3}{25} )</td>
</tr>
</tbody>
</table>

**A2.3 Stiffness and Solution Stability**

In most practical applications, coupled processes occur at different rates. Depending on whether the rates of change differ slightly or significantly, there may be multiple scales of behaviour. Consider an IVP with two variables, \( y_1 \) and \( y_2 \), where each decays to zero from some initial value. If \( |\frac{dy_1}{dt}| \) and \( |\frac{dy_2}{dt}| \) are of similar size, then the two variables will evolve on the same temporal scale; however, if \( |\frac{dy_1}{dt}| \) and \( |\frac{dy_2}{dt}| \) differ substantially, then the solution will comprise two scales of behaviour, one small and the other large. Even scalar problems may exhibit this type of behaviour. Such differential equations are often difficult to numerically integrate, which gives rise to the concept of *stiffness*. A stiff differential equation (or system of stiff differential equations) is one that is difficult to solve numerically due to the presence of largely varying rates of change.

When solving stiff equations, the maximum permissible \( \Delta t \) is dictated by stability requirements rather than those for accuracy. If too large of a \( \Delta t \) is selected, not only will the solution fail to capture the small-scale changes, but it will become unstable. Studying stiff systems of equations is thus of interest, for the numerical method (see §A2.3) and \( \Delta t \) must be selected to ensure both an accurate and stable solution. The concept of stiffness is explored below, along with the numerical solution of stiff systems of equations.

**A2.3.1 Problem Definition**

There are multiple definitions of stiffness, none of which are universally accepted. Some of the more common ones are listed below:

- Stability dictates the required integration interval as opposed to the desired accuracy.
Appendix A: Review of Numerical Methods

- Systems comprising different scales of behaviour are often stiff.
- A system of differential equations is stiff if the real parts of the eigenvalues of the system’s Jacobian are all negative and if the stiffness ratio is large.

Definition no. 1 broadly defines stiffness. In fact, this is the key characteristic of a stiff differential equation. With respect to the second, differing rates of decay commonly arise when solving for the constituents within multi-step chemical reactions. If too large a $\Delta t$ is used, the smaller-scale reaction rates will not be resolved adequately; as a result, the numerical solution may become unstable. The third definition is similar to the first, but it is better founded numerically. The relationship between eigenvalues and stiffness is discussed below.

A2.3.2 Eigenvalues and Stiffness

Recall from §A1.2 that the eigenvalues of a matrix $A$, denoted by $\sigma(A) \in \mathbb{C}^n$, describe its transformative action when applied to a vector. For a system of ODEs such as Equation [A35], the eigenvalues of the Jacobian $J = \nabla f \in \mathbb{R}^{n \times n}$ (i.e., the gradient of $f$ with respect to $y$) characterize the way in which the numerical approximation of $y(t)$ evolves over time. If the ODE is linear, then $J = A$; otherwise, the gradient must be evaluated.

The sign, real and imaginary components, and magnitude of each eigenvalue all play a role. The real components of each $\lambda$ relate to the rates at which the solution changes, and the imaginary components indicate whether there are oscillations. If the real part of a system’s eigenvalues are negative, then the system is progressing towards some steady state. Comparatively, positive values for the real components signify that a system has no equilibrium. Lastly, a system’s rates of change are related to the magnitude of its eigenvalues, denoted by $|\lambda_i|$: small $|\lambda_i|$ imply small rates of change, while large $|\lambda_i|$ indicate larger rates of change.

Further to the above, the presence of largely varying orders of $|\lambda_i|$ indicates that a system of ODEs is stiff. That is, when $|\lambda_i|$ varies substantially, both small-scale changes and long-term behaviour must be resolved in the numerical solution. One means of characterizing the stiffness of a system of ODEs is through its stiffness ratio

$$S = \frac{\max|\text{Re}(\lambda)|}{\min|\text{Re}(\lambda)|} \quad [A45]$$

which is similar to the definition of $\kappa(A)$ of a matrix. Large values for $S$ imply significantly varying rates of change and thus stiffness. When $S$ is large (and a system of ODEs is stiff), it becomes difficult to obtain a stable numerical solution, and the key to resolving this is the choice...
of a suitable numerical method and integration interval.

**A2.3.3 Numerical Stability**

In addition to numerical accuracy, different numerical methods have different stability characteristics. For stiff systems of ODEs, methods are required that yield an accurate and stable solution; thus, it is of interest to characterize a method’s stability region. This allows different methods to be compared on the basis of their stability properties.

The stability of a numerical method is typically characterized by the region in which \( \lim_{t \to \infty} y(t) = 0 \) for the scalar IVP

\[
\frac{dy}{dt} = a \cdot y
\]

where \( y(0) = 1 \) and \( \text{Re}(a) < 0 \). Note that \( a \) may be complex, but here \( a \in \mathbb{R} \) is assumed. Because this IVP describes exponential decay, the numerical solution should also decay asymptotically; accordingly, a numerical method will be stable for this problem if \( |\phi(z)| < 1 \), where \( z = a \cdot \Delta t \), and the stability function \( \phi(z) = y^{(n+1)}/y^{(n)} \). If \( |\phi(z)| < 1 \) for all \( z \), a numerical method is said to be asymptotically stable or A-stable.

To illustrate the concept of stability regions, consider the first order Euler and backward Euler methods. For the former (from Equation [A36]), the numerical recursion is given as

\[
y^{(n+1)} = (z + 1)y^{(n)}
\]

and for the backward Euler method (from Equation [A40]), the numerical recursion is

\[
y^{(n+1)} = \left(\frac{1}{1-z}\right)y^{(n)}
\]

The stability functions for the Euler and backward Euler methods are thus \( \phi(z) = z + 1 \) and \( \phi(z) = 1/(1 - z) \), respectively. A numerical solution will be stable if \( |\phi(z)| < 1 \), so the Euler method’s stability region is bound by \(-2 < z < 0\) (or, more conveniently, \( \Delta t < |2/a| \)). Conversely, \( |\phi(z)| < 1 \) for all \( z \) for the backward Euler method. This is because \( \phi(z) = 1/(1 - z) \) is always negative, which results from \( z = a \cdot \Delta t < 0 \) for \( a < 0 \) (by the problem definition). The backward Euler method is thus A-stable.

**A2.3.4 Stiffness and Solution Stability**

Two key concepts have been presented thus far for stiff systems of equations. The stiffness
of a system of ODEs is characterized by its stiffness ratio, and the key to obtaining a stable numerical solution is the selection of an appropriate numerical method and $\Delta t$. Both concepts are illustrated below via an example.

Consider a linear IVP (Equation [A35]) with conditions

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -2k \end{bmatrix}, \quad y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}, \quad \phi(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$ \quad [A49]

where $k > 1$ is a parameter. Equation [A49] describes the interdependent exponential decay of $y_1(t)$ and $y_2(t)$ from their respective initial values. Note that because the IVP is linear, its Jacobian is simply the coefficient matrix $A$. Furthermore, because $A$ is negative-definite for all $k > 1$, its eigenvalues will always be negative real numbers.

First we will compute the eigenpairs of $A$. The eigenvalues are determined by solving for the roots of the characteristic equation

$$\det(A) = \begin{vmatrix} -2 - \lambda & 1 \\ 1 & -2k - \lambda \end{vmatrix} = 0$$ \quad [A50]

which leads to

$$\lambda_1 = -(k + 1) + \sqrt{(k - 1)^2 + 1}$$
$$\lambda_2 = -(k + 1) - \sqrt{(k - 1)^2 + 1}$$ \quad [A51]

The eigenvectors $E_1$ and $E_2 \in \mathbb{R}^n$ corresponding to $\lambda_1$ and $\lambda_2$ can be obtained from

$$\begin{bmatrix} -2 - \lambda_1 & 1 \\ 1 & -2k - \lambda_1 \end{bmatrix} E = 0$$ \quad [A52]

By solving Equation [A52], the eigenvectors are

$$E = [E_1 \quad E_2] = \begin{bmatrix} 1 \\ \lambda_1 + 2 \quad \lambda_2 + 2k \end{bmatrix}$$ \quad [A53]

Using the complete set of eigenpairs, an expression for the exact solution, derived through analytical means, can be assembled. The exact solution to the IVP is

$$y(t) = c_1 e^{\lambda_1 t} E_1 + c_2 e^{\lambda_2 t} E_2, \quad \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = E^{-1} y_0$$ \quad [A54]

With the exact solution known, the performance of different numerical solutions can be studied. From §A2.3.2, the stiffness ratio $S$ characterizes the stiffness of a system of differential equations. With Equation [A51] and $k > 1$, the stiffness ratio for the ODE IVP is
Appendix A: Review of Numerical Methods

\[ S = \frac{k+1+\sqrt{(k-1)^2+1}}{k+1-\sqrt{(k-1)^2+1}} \quad [A55] \]

From Equation [A55], increasing \( k \) results in greater \( S \) and a stiffer system of differential equations; in fact, \( S \) converges to \( k \) as \( k \) becomes large. Thus, the maximum permissible \( \Delta t \) to ensure a stable solution is approximately inversely proportional to \( k \).

The Euler and backward Euler methods will be used to solve the IVP with \( k = 100 \). To obtain a stable solution, a sufficiently small \( \Delta t \) is needed: for now, let us use \( \Delta t = 1/nk \), where \( n \) is an integer. Numerical solutions for various \( n \) are illustrated on the following page (Figure A2). A comparison of the results shows that the solutions from the Euler method fall below the exact solution, while those from the backward Euler method plot above it: this is because the former is explicit and the latter is implicit. Additionally, the results for the Euler method become unstable as \( n \to 1 \), a result of the numerical recursion overshooting the initial integration step by more and more. Comparatively, the implicit backward Euler method is stable for all \( n \), including \( n < 1 \).

A2.4 Summary

The numerical solution of systems of differential equations can be performed using one of many numerical methods. In selecting one, along with \( \Delta t \), consideration must be given to both the numerical accuracy and the numerical stability of the solution; to do otherwise would give the numerical results little meaning. The emphasis of this section has been exactly that. As with the different methods for solving systems of linear equations, the selection of a numerical method and \( \Delta t \) should reflect the underlying numerical problem. If a solution of greater accuracy is required, then the numerical method and \( \Delta t \) should be chosen with this in mind, and, in the special case of stiff systems of differential equations, additional care is needed.

A3. Brief Remarks on Pipe Network Analysis

Throughout this appendix discussions have been numerical in nature. First, solution methods for systems of linear and non-linear equations were reviewed, including direct and iterative methods, pivoting, preconditioning, saddle point problems, and sparse matrices. Following this, we explored the numerical solution of systems of linear ODE IVPs. Central to both topics are numerical accuracy and numerical stability, each of which is necessary. Although treated in separate sections, many applications also involve both the solution of systems of equations and the numerical integration of ODE IVPs.
Appendix A: Review of Numerical Methods

With respect to this thesis, all of the aforementioned are relevant. In modeling the unsteady hydraulics of pipe networks, large sparse systems of differential equations arise: they describe the fundamental conservation laws of mass, momentum, and energy for every node and link in a network. In solving these, the selection and implementation of suitable numerical methods is key. For discussions of the specific roles of the concepts overviewed in the preceding sections, the reader is referred to Chapter 2 and 3 of this thesis. Altogether, the effective and efficient solution of the governing equations of unsteady pipe network hydraulics requires an understanding of both the numerical solution of systems of equations and ODE IVPs.
Appendix A: Review of Numerical Methods

(a) Solution for $n = 5$

(b) Solution for $n = 3$

(c) Solution for $n = 2$

(d) Solution for $n = 1.5$

(e) Solution for $n = 1.25$

(f) Solution for $n = 1.125$

Figure A2: Solutions to Equation [A49] for various $n$
Appendix A: Review of Numerical Methods

References


Appendix B: Generalized Characteristic Method Implementation Algorithms

Appendix B contains supplemental information for the generalized characteristic method (GCM) presented in Chapter 6. In addition to the compatibility equations and solutions, implementation algorithms are presented for both interior and junction analyses. The base algorithm represents the GCM in its general form, that is, with the two-coefficient ($\theta$ and $\varepsilon$) friction integral approximation, the variable reach scheme, and unsteady friction based on the single coefficient modified instantaneous-based acceleration model. From this, reduced implementations are presented for quasi-friction friction, steady friction, and explicit values for $\theta$ and $\varepsilon$: the purpose of this is to ascertain how the total number of floating point operations (FLOPs), and thus computational effort, varies with the specific implementation. Note that one FLOP comprises a single addition, subtraction, multiplication, or division operation. The associated number of FLOPs for each computation within a given implementation are shown to the right of the computation in braces, for example, (6) denotes 6 FLOPs.

B1. Generalized Characteristic Method

B1.1 Compatibility Expressions and Solutions

Positive compatibility equation for interior analysis:

$$B_A Q_P + H_P - C_A = 0 \quad \text{[B1]}$$

$$B_A = \left(1 + \frac{1}{2} k\right) B + \theta K_P |Q_P| + (1 - \theta) \varepsilon K_A |Q_A|$$

$$C_A = (B - (1 - \theta)(1 - \varepsilon)K_A|Q_A|)Q_A + H_A + F_A$$

$$F_A = \frac{1}{2} B k \left(Q_P^{(t-2r)} - \text{sign}(Q_A)|Q_B - Q_A|\right)$$

Negative compatibility equation for interior analysis:

$$B_B Q_P - H_P - C_B = 0 \quad \text{[B2]}$$

$$B_B = \left(1 + \frac{1}{2} k\right) B + \theta K_P |Q_P| + (1 - \theta) \varepsilon K_B |Q_B|$$

$$C_B = (B - (1 - \theta)(1 - \varepsilon)K_B|Q_B|)Q_B - H_B + F_B$$
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ F_B = \frac{1}{2} B k \left( Q_P^{(t-2r)} - \text{sign}(Q_B)|Q_B - Q_A| \right) \]

Positive compatibility equation for junction analysis:

\[ B_A Q_P + H_P - C_A = 0 \]  \[ \text{[B3]} \]
\[ B_A = (1 + k)B + \theta K_P|Q_P| + (1 - \theta)\varepsilon K_A|Q_A| \]
\[ C_A = (B - (1 - \theta)(1 - \varepsilon)K_A|Q_A|)Q_A + H_A + F_A \]
\[ F_A = B k \left( Q_P^{(t-r)} - \text{sign}(Q_A)|Q_P^{(t-r)} - Q_A| \right) \]

Negative compatibility equation for junction analysis:

\[ B_B Q_P - H_P - C_B = 0 \]  \[ \text{[B4]} \]
\[ B_B = (1 + k)B + \theta K_P|Q_P| + (1 - \theta)\varepsilon K_B|Q_B| \]
\[ C_B = (B - (1 - \theta)(1 - \varepsilon)K_B|Q_B|)Q_B - H_B + F_B \]
\[ F_B = B k \left( Q_P^{(t-r)} - \text{sign}(Q_B)|Q_B - Q_P^{(t-r)}| \right) \]

Interior analysis solution:

\[ J^{(m)} Q_P^{(m+1)} = 2\theta K_P^{(m)} \left| Q_P^{(m)} \right| Q_P^{(m)} + C_A + C_B \]  \[ \text{[B5]} \]
\[ J^{(m)} = 4\theta K_P^{(m)} \left| Q_P^{(m)} \right| + (2 + k)B + (1 - \theta)\varepsilon (K_A|Q_A| + K_B|Q_B|) \]
\[ H_P = C_A - B_A Q_P \]  \[ \text{[B6]} \]

Junction analysis solution:

\[ H_P = \left( \sum_A \frac{1}{(B_A)_i} + \sum_B \frac{1}{(B_B)_i} \right)^{-1} \left( \sum_A \frac{(C_A)_i}{(B_A)_i} - \sum_B \frac{(C_B)_i}{(B_B)_i} - q_P \right) \]  \[ \text{[B7]} \]

**B1.2 Interior Analysis Implementation**

% Calculate static constants [19 FLOPs].

\[ B = a/\gamma A \]  \[ (2) \]
\[ r = N_0/N_R \]  \[ (1) \]
\[ \Delta x_j = L_j/N_{Rj} \]  \[ (1) \]
\[ C_1 = D/\nu A \]  \[ (2) \]
\[ C_2 = \varepsilon/3.7D \]  \[ (2) \]
\[ C_3 = \Delta x/8gDA^2 \]  \[ (5) \]
\[ C_4 = 1 - \theta \]  \[ (1) \]
\[ C_5 = C_4(1 - \varepsilon) \]  \[ (2) \]
\[ C_6 = C_4\varepsilon \]  \[ (1) \]
Appendix B: Generalized Characteristic Method  
Implementation Algorithms

\[ C_B = 2 \theta \]  \hspace{1cm} (1)
\[ C_0 = B/2 \]  \hspace{1cm} (1)

For \( j = 1: N_p \) \% Loop through pipes.

For \( i = 2: N_R - 1 \) \% Loop through interior nodes.

\% Calculate dynamic constants \[47 \text{ FLOPs/reach}\].
\[ Re_A = C_1 |Q_A| \]  \hspace{1cm} (1)
\[ Re_B = C_1 |Q_B| \]  \hspace{1cm} (1)
\[ Re_P = C_1 |Q_{P-r}'| \]  \hspace{1cm} (1)

\[ K_A = \begin{cases} 
C_3 \left( \log_{10} \left( C_2 + \frac{5.74}{Re_A^{0.9}} \right) \right)^{-2} & \text{Turbulent} \\
256C_3/Re_A & \text{Laminar}
\end{cases} \]  \hspace{1cm} (6)
\[ K_B = \begin{cases} 
C_3 \left( \log_{10} \left( C_2 + \frac{5.74}{Re_B^{0.9}} \right) \right)^{-2} & Re_B > 2 \times 10^3 \\
256C_3/Re_B & Re_B \leq 2 \times 10^3
\end{cases} \]  \hspace{1cm} (6)

\[ k = \begin{cases} 
\left( 1.853 \times Re_p^{- \log_{10}(14.3 \times Re_p^{0.05})} \right)^{1/2} & Re_P > 2 \times 10^3 \\
4.76 \times 10^{-3} & Re_P \leq 2 \times 10^3
\end{cases} \]  \hspace{1cm} (6)

\[ C_{10} = C_9 k \]  \hspace{1cm} (1)
\[ C_{11} = |Q_B - Q_A| \]  \hspace{1cm} (1)
\[ F_A = C_{10}(Q_{P-r}' - \text{sign}(Q_A)C_{11}) \]  \hspace{1cm} (3)
\[ F_A = C_{10}(Q_{P-r}' - \text{sign}(Q_A)C_{11}) \]  \hspace{1cm} (3)
\[ C_{12} = K_A |Q_A| \]  \hspace{1cm} (1)
\[ C_{13} = K_B |Q_B| \]  \hspace{1cm} (1)
\[ C_A = (B - C_5C_{12})Q_A + H_A + F_A \]  \hspace{1cm} (5)
\[ C_B = (B - C_5C_{13})Q_B - H_B + F_B \]  \hspace{1cm} (5)
\[ C_{14} = (2 + k)B + C_6(C_{12} + C_{13}) \]  \hspace{1cm} (5)
\[ C_{15} = C_A + C_B \]  \hspace{1cm} (1)

\% Initialize flow estimate.
\[ Q_P^0 = Q_P^b \]  \hspace{1cm} (1)

\% Calculate flow solution \[16 \text{ FLOPs/iteration-reach}].
While true
\[ Q_P^b = Q_P^1 \]
\[ Re_P = C_1 |Q_P^b| \]  \hspace{1cm} (1)
\[ K_P = \begin{cases} 
C_3 \left( \log_{10} \left( C_2 + \frac{5.74}{Re_P^{0.9}} \right) \right)^{-2} & \text{Turbulent} \\
256C_3/Re_P & \text{Laminar}
\end{cases} \]  \hspace{1cm} (6)
\[ C_{16} = C_8 K_P |Q_P^b| \]  \hspace{1cm} (2)
\[ C_{17} = 2C_{16} \]  \hspace{1cm} (1)
\[ J = C_{17} + C_{14} \]  \hspace{1cm} (1)
Appendix B: Generalized Characteristic Method Implementation Algorithms

\[ Q_1^p = (C_{16}Q_0^p + C_{15})/J \] (3)

If \( \theta = 0 \) or \( |Q_p^b - Q_0^p|/|Q_p^1| < e_{tol} \), break

\[ B_A = B + C_{10} + \theta K_P |Q_p^b| + C_6 C_{12} \] (6)
\[ H_p = C_A - B_A Q_p^b \] (2)

### B1.3 Junction Analysis Implementation

% Calculate static constants [13 FLOPs/pipe].

\[ B_j = a_j/gA_j \] (2)
\[ r_j = N_{0j}/N_{Rj} \] (1)
\[ \Delta x_j = L_j/N_{Rj} \] (1)
\[ C_{1j} = D_j/vA_j \] (2)
\[ C_{2j} = \varepsilon_j/3.7D_j \] (2)
\[ C_{3j} = \Delta x_j/8gD_jA_j^2 \] (5)

% Junction analysis for each time step.

For \( i = 1: N_j \) % Loop through junctions.

% Initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]
\[ f = 0 \]

% Calculate dynamic constants for pipes adjacent to junction \( i \) [32 FLOPs/pipe].

For \( j \in [N_{Bi}, N_{Ai}] \)

\[ Q_{Rj} = \begin{cases} Q_{(j,N_{Rj})}^{t-r_j} & j \in N_{Bi} \\ Q_{(j,2)}^{t-r_j} & j \in N_{Ai} \end{cases} \]
\[ H_R = \begin{cases} H_{(j,N_{Rj})}^{t-r_j} & j \in N_{Bi} \\ -H_{(j,2)}^{t-r_j} & j \in N_{Ai} \end{cases} \]
\[ Q_{P}^{t-r} = \begin{cases} Q_{(j,N_{Rj}+1)}^{t-r_j} & j \in N_{Bi} \\ Q_{(j,1)}^{t-r_j} & j \in N_{Ai} \end{cases} \]
\[ R_{eR} = C_{1j} |Q_{Rj}| \] (1)
\[ R_{ep} = C_{1j} |Q_{P}^{t-r}| \] (1)
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ K_{Rj} = \begin{cases} 
C_3 R_j \left( \log_{10} \left( C_2 R_j + \frac{5.74}{Re_R^{0.9}} \right) \right)^{-2} & \text{Turbulent} \\
256 C_3 R_j / Re_R & \text{Laminar}
\end{cases} \]  \hspace{1cm} (6)

\[ C_5 = K_{Rj} \left| Q_{Rj} \right| \]  \hspace{1cm} (1)

\[ C_{6j} = \varepsilon_j C_5 \]  \hspace{1cm} (1)

\[ C_{7j} = C_5 - C_{6j} \]  \hspace{1cm} (1)

\[ k = \begin{cases} 
\left( 1.853 \times Re_R^{-1} \log_{10} \left( 14.3 \times Re_R^{-0.05} \right) \right)^{1/2} & \text{Turbulent} \\
4.76 \times 10^{-3} & \text{Laminar}
\end{cases} \]  \hspace{1cm} (6)

\[ C_8 = Bk \]  \hspace{1cm} (1)

\[ F = C_8 \left( Q_{t-r}^P - \text{sign}(Q_{t-r}^P)\left| Q_{t-r}^P - Q_{t-r}^P \right| \right) \]  \hspace{1cm} (4)

\[ B_{Rj} = B_j + C_{6j} + C_8 \]  \hspace{1cm} (2)

\[ C_{9j} = 1 / B_{Rj} \]  \hspace{1cm} (1)

\[ B_0 = B_0 + C_{9j} \]  \hspace{1cm} (1)

\[ C_{Rj} = \left( B_j - C_{7j} \right) Q_{Rj} + H_R + F \]  \hspace{1cm} (4)

\[ C_{10j} = C_{Rj} / B_{Rj} \]  \hspace{1cm} (1)

\[ C_0 = \begin{cases} 
C_0 + C_{10j} & j \in N_{B_i} \\
C_0 - C_{10j} & j \in N_{A_i}
\end{cases} \]  \hspace{1cm} (1)

\[ \text{If } \theta_j > 0, f = 1 \]

\% Calculate predictor head solution [1 FLOP].
\[ H_i = C_0 / B_0 \]  \hspace{1cm} (1)

\% Calculate predictor flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ Q_{Pj} = \begin{cases} 
\left( C_{Rj} - H_i \right) / B_{Rj} & j \in N_{B_i} \\
\left( C_{Rj} + H_i \right) / B_{Rj} & j \in N_{A_i}
\end{cases} \]  \hspace{1cm} (2)

\% Update dynamic constants for pipes adjacent to junction \( i \).
If \( f = 1 \)
\% Re-initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]

\% Update dynamic constants [18 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ \text{If } \theta_j > 0 \]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ Q_R = \begin{cases} Q_{(j,N_{R_j})}^{t-r_j} & j \in N_{B_i} \\ Q_{(j,2)}^{t-r_j} & j \in N_{A_i} \end{cases} \]

\[ R e_P = C_1 \left| Q_P \right| \]  

\[ K_{P_j} = \begin{cases} C_3 \left( \log_{10} \left( C_2 + \frac{5.74}{R e_P^{0.65}} \right) \right)^{-2} & \text{Turbulent} \\ 256C_3/R e_P & \text{Laminar} \end{cases} \]

\[ B_{Rj} = B_{Rj} + \theta_j \left( K_{P_j} \left| Q_{P_j} \right| - C_{6j} \right) \]  

\[ C_{9j} = 1/B_{Rj} \]  

\[ C_{Rj} = C_{Rj} + \theta_j C_7 Q_{Rj} \]  

\[ C_{10j} = C_{Rj}/B_{Rj} \]  

\[ B_0 = B_0 + C_{9j} \]  

\[ C_0 = \begin{cases} C_0 + C_{10j} & j \in N_{B_i} \\ C_0 - C_{10j} & j \in N_{A_i} \end{cases} \]  

\% Calculate corrector head solution [1 FLOP].
\[ H_i = C_0/B_0 \]

\% Calculate corrector flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)

\[ Q_{P_j} = \begin{cases} \left( C_{Rj} - H_i \right)/B_{Rj} & j \in N_{B_i} \\ \left( C_{Rj} + H_i \right)/B_{Rj} & j \in N_{A_i} \end{cases} \]  

\[ B_{2}. \text{GCM with Quasi-Steady Friction} \]

\[ B_{2.1} \text{ Compatibility Expressions and Solutions} \]

Positive compatibility equation for interior and junction analyses:

\[ B_A Q_P + H_P - C_A = 0 \]  

\[ B_A = B + \theta K_P |Q_P| + (1 - \theta) \epsilon K_A |Q_A| \]

\[ C_A = (B - (1 - \theta)(1 - \epsilon) K_A |Q_A|) Q_A + H_A \]

Negative compatibility equation for interior and junction analyses:

\[ B_B Q_P - H_P - C_B = 0 \]  

\[ B_B = B + \theta K_P |Q_P| + (1 - \theta) \epsilon K_B |Q_B| \]

\[ C_B = (B - (1 - \theta)(1 - \epsilon) K_B |Q_B|) Q_B - H_B \]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

Interior analysis solution:

\[
J^{(m)}Q_p^{(m+1)} = 2\theta K_p^{(m)}\left|Q_p^{(m)}\right|Q_p^{(m)} + C_A + C_B \tag{B10}
\]

\[
J^{(m)} = 4\theta K_p^{(m)}\left|Q_p^{(m)}\right| + 2B + (1 - \theta)\varepsilon(K_A|Q_A| + K_B|Q_B|)
\]

\[
H_p = C_A - B_A Q_p \tag{B11}
\]

Junction analysis solution:

\[
H_p = \left(\sum_{(B_A)} \frac{1}{C_A} + \sum_{(B_B)} \frac{1}{C_B}\right)^{-1} \left(\sum_{(B_A)} \frac{C_A}{C_A} - \sum_{(B_B)} \frac{C_B}{C_B} - q_p \right) \tag{B12}
\]

### B2.2 Interior Analysis Implementation

% Pre-calculated static constants [18 FLOPs].
\[
B = a/gA \tag{2}
\]
\[
\Delta x_j = L_j/N_{R_j} \tag{1}
\]
\[
C_1 = D/\nu A \tag{2}
\]
\[
C_2 = \varepsilon/3.7D \tag{2}
\]
\[
C_3 = \Delta x/8gDA^2 \tag{5}
\]
\[
C_4 = 1 - \theta \tag{1}
\]
\[
C_5 = C_4(1 - \varepsilon) \tag{2}
\]
\[
C_6 = C_4\varepsilon \tag{1}
\]
\[
C_7 = 2B \tag{1}
\]
\[
C_9 = 2\theta \tag{1}
\]

For \( j = 1: N_p \) % Loop through pipes.

For \( i = 2: N_{R_i} - 1 \) % Loop through interior nodes.

% Calculate dynamic constants [28 FLOPs/each].
\[
Re_A = C_1|Q_A| \tag{1}
\]
\[
Re_B = C_1|Q_B| \tag{1}
\]
\[
K_A = \begin{cases} 
C_3 \left(\log_{10} \left(\frac{C_2 + \frac{5.74}{Re_A^{0.9}}}{256C_3/Re_A} \right)\right)^{-2} & Re_A > 2 \times 10^3 \\
256C_3/Re_A & Re_A \leq 2 \times 10^3 
\end{cases} \tag{6}
\]
\[
K_B = \begin{cases} 
C_3 \left(\log_{10} \left(\frac{C_2 + \frac{5.74}{Re_B^{0.9}}}{256C_3/Re_B} \right)\right)^{-2} & Re_B > 2 \times 10^3 \\
256C_3/Re_B & Re_B \leq 2 \times 10^3 
\end{cases} \tag{6}
\]
\[
C_{11} = K_A|Q_A| \tag{1}
\]
\[
C_{12} = K_B|Q_B| \tag{1}
\]
\[
C_A = (B - C_5C_{11})Q_A + H_A \tag{4}
\]
\[
C_B = (B - C_5C_{12})Q_B - H_B \tag{4}
\]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ C_{13} = C_7 + C_6(C_{11} + C_{12}) \]  \hspace{1cm} (3)
\[ C_{14} = C_A + C_B \]  \hspace{1cm} (1)

% Initialize flow estimate.
\[ Q_P^1 = Q^t_P \]

% Calculate flow solution [17 FLOPs/iteration-reach].
While true
\[ Q_P^0 = Q_P^1 \]
\[ Re_P = C_1 |Q_P^0| \]  \hspace{1cm} (1)
\[ K_P = \begin{cases} 
   C_3 \left( \log_{10} \left( C_2 + \frac{5.74}{Re_P} \right) \right)^{-2} & Re_P > 2 \times 10^3 \\
   256C_3/Re_P & Re_P \leq 2 \times 10^3 
\end{cases} \]
\[ C_{15} = C_9 K_P |Q_P^0| \]  \hspace{1cm} (2)
\[ C_{16} = 2C_{15} \]  \hspace{1cm} (1)
\[ J^m = C_{16} + C_{13} \]  \hspace{1cm} (1)
\[ Q_P^1 = ((C_{16} - C_{15})Q_P^0 + C_{14})/J \]  \hspace{1cm} (4)
If \( \theta = 0 \) or \( |Q_P^1 - Q_P^0|/|Q_P^1| < e_{tol} \), break

% Calculate head solution [7 FLOPs/reach].
\[ B_A = B + \theta K_P |Q_P^1| + C_6 C_{11} \]  \hspace{1cm} (5)
\[ H_P = C_A - B_A Q_P^1 \]  \hspace{1cm} (2)

\section*{B2.3 Junction Analysis Implementation}

% Calculate static constants [13 FLOPs/pipe].
\[ B_j = a_j/gA_j \]  \hspace{1cm} (2)
\[ r_j = N_{0j}/N_{Rj} \]  \hspace{1cm} (1)
\[ \Delta x_j = L_j/N_{Rj} \]  \hspace{1cm} (1)
\[ C_{1j} = D_j/vA_j \]  \hspace{1cm} (2)
\[ C_{2j} = \epsilon_j/3.7D_j \]  \hspace{1cm} (2)
\[ C_{3j} = \Delta x_j/8gD_jA_j^2 \]  \hspace{1cm} (5)

% Junction analysis for each time step.
For \( i = 1: N_j \) % Loop through junctions.
% Initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]
\[ f = 0 \]
% Calculate dynamic constants for pipes adjacent to junction $i$ [18 FLOPs/pipe].

For $j \in [N_{B_i}, N_{A_i}]$

$$Q_{R_j} = \begin{cases} Q_{(j,N_{R_j})}^{t-r_j} & j \in N_{B_i} \\ Q_{(j,2)}^{t-r_j} & j \in N_{A_i} \end{cases}$$

$$H_R = \begin{cases} H_{(j,N_{R_j})}^{t-r_j} & j \in N_{B_i} \\ -H_{(j,2)}^{t-r_j} & j \in N_{A_i} \end{cases}$$

$$Re_R = C_1 j |Q_{R_j}|$$

$$K_{R_j} = \begin{cases} C_3 j \left( \log_{10} \left( C_{2j} + \frac{5.74}{Re_{0.9}} \right) \right)^{-2} & Re_R > 2 \times 10^3 \\ 256 C_3 j / Re_R & Re_R \leq 2 \times 10^3 \end{cases}$$

$$C_6 = K_{R_j} |Q_{R_j}|$$

$$C_7_j = \varepsilon_j C_6$$

$$C_8_j = C_6 - C_7_j$$

$$B_{R_j} = B_j + C_7_j$$

$$C_{10_j} = 1 / B_{R_j}$$

$$B_0 = B_0 + C_{10_j}$$

$$C_{R_j} = (B_j - C_8_j) Q_{R_j} + H_R$$

$$C_{11_j} = C_{R_j} / B_{R_j}$$

$$C_0 = \begin{cases} C_0 + C_{11_j} & j \in N_{B_i} \\ C_0 - C_{11_j} & j \in N_{A_i} \end{cases}$$

If $\theta_j > 0$, $f = 1$

% Calculate predictor head solution [1 FLOP].

$$H_i = C_0 / B_0$$

% Calculate predictor flow solutions [2 FLOPs/pipe].

For $j \in [N_{B_i}, N_{A_i}]$

$$Q_{P_j} = \begin{cases} \left( C_{R_j} - H_i \right) / B_{R_j} & j \in N_{B_i} \\ \left( C_{R_j} + H_i \right) / B_{R_j} & j \in N_{A_i} \end{cases}$$

% Update dynamic constants for pipes adjacent to junction $i$.

If $f = 1$

% Re-initialize counters.

$B_0 = 0$
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ C_0 = -q_i \]

\% Update dynamic constants [18 FLOPs/pipe].
For \( j \in [N_{Bi}, N_{Ai}] \)
If \( \theta_j > 0 \)
\[
Q_R = \begin{cases} 
Q_{(j,N_{Rj})}^{t-r_j} & j \in N_{Bi} \\
Q_{(j,2)}^{t-r_j} & j \in N_{Ai} 
\end{cases}
\]

\[ Re_p = C_1 \left| Q_{Pj} \right| \]

\[ K_{Pj} = \begin{cases} 
C_3 \left( \log_{10} \left( C_2 + \frac{5.74}{Re_p^{0.5}} \right) \right)^2 & Re_p > 2 \times 10^3 \\
256C_3/Re_p & Re_p \leq 2 \times 10^3 
\end{cases} \]

\[ B_{Rj} = B_{Rj} + \theta_j \left( K_{Pj} \left| Q_{Pj} \right| - C_7 \right) \]

\[ C_{10j} = 1/B_{Rj} \]

\[ C_{Rj} = C_{Rj} + \theta_j C_{8j} Q_{Rj} \]

\[ C_{11j} = C_{Rj}/B_{Rj} \]

\[ B_0 = B_0 + C_{10j} \]

\[ C_0 = \begin{cases} 
C_0 + C_{11j} & j \in N_{Bi} \\
C_0 - C_{11j} & j \in N_{Ai} 
\end{cases} \]

\% Calculate corrector head solution [1 FLOP].
\[ H_i = C_0/B_0 \]

\% Calculate corrector flow solutions [2 FLOPs/pipe].
For \( j \in [N_{Bi}, N_{Ai}] \)
\[
Q_{Pj} = \begin{cases} 
(C_{Rj} - H_i) / B_{Rj} & j \in N_{Bi} \\
(C_{Rj} + H_i) / B_{Rj} & j \in N_{Ai} 
\end{cases} \]

B3. GCM with Steady Friction

B3.1 Compatibility Expressions and Solutions

Positive compatibility equation for interior and junction analyses:

\[ B_A Q_P + H_P - C_A = 0 \] \[ \text{[B13]} \]

\[ B_A = B + K(\theta |Q_P| + (1 - \theta)\varepsilon |Q_A|) \]

\[ C_A = (B - (1 - \theta)(1 - \varepsilon)K|Q_A|)Q_A + H_A \]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

Negative compatibility equation for interior and junction analyses:

\[ B_B Q_P - H_P - C_B = 0 \] \[ B_B = B + K(\theta |Q_P| + (1 - \theta)\varepsilon |Q_B|) \]
\[ C_B = (B - (1 - \theta)(1 - \varepsilon)K|Q_B|)Q_B - H_B \]

Interior analysis solution:

\[ J^{(m)} Q_p^{(m+1)} = 2\theta K|Q_p^{(m)}| + C_A + C_B \] \[ J^{(m)} = 4\theta K|Q_p^{(m)}| + 2B + (1 - \theta)\varepsilon K(|Q_A| + |Q_B|) \]
\[ H_P = C_A - B_A Q_P \]

Junction analysis solution:

\[ H_P = \left( \sum_A \frac{1}{|Q_A|} + \sum_B \frac{1}{|Q_B|} \right)^{-1} \left( \sum_A \frac{|C_A|}{|Q_A|} - \sum_B \frac{|C_B|}{|Q_B|} - Q_P \right) \]

B3.2 Interior Analysis Implementation

% Pre-calculated static constants [15 FLOPs].
\[ B = a/gA \] \[ \Delta x_j = L_j/N_R \] \[ K = f\Delta x/8gDA^2 \] \[ C_4 = 1 - \theta \] \[ C_5 = C_4(1 - \varepsilon) \] \[ C_6 = C_4\varepsilon \] \[ C_7 = 2B \] \[ C_9 = 2\theta \]

For \( j = 1: N_P \) % Loop through pipes.
\[ C_{11} = K|Q_A| \] \[ C_{12} = K|Q_B| \] \[ C_A = (B - C_5 C_{11})Q_A + H_A \] \[ C_B = (B - C_5 C_{12})Q_B - H_B \] \[ C_{13} = C_7 + C_6(C_{11} + C_{12}) \] \[ C_{14} = C_A + C_B \]

% Calculate dynamic constants [14 FLOPs/branch].

For \( i = 2: N_R - 1 \) % Loop through interior nodes.
% Initialize flow estimate.
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ Q_P^1 = Q_P^t \]

% Calculate flow solution [10 FLOPs/iteration-reach].
While true
\[ Q_P^0 = Q_P^1 \]
\[ C_{15} = C_9 K |Q_P^0| \]
\[ C_{16} = 2 C_{15} \]
\[ f_m = C_{16} + C_{13} \]
\[ Q_P^1 = ((C_{16} - C_{15}) Q_P^0 + C_{14}) / J \]
If \( \theta = 0 \) or \(|Q_P^1 - Q_P^0|/|Q_P^1| < e_{tol} \), break

% Calculate head solution [7 FLOPs/reach].
\[ B_A = B + \theta K |Q_P^1| + C_6 C_{11} \]
\[ H_P = C_A - B_A Q_P^1 \]

B3.3 Junction Analysis Implementation

% Calculate static constants [10 FLOPs/pipe].
\[ B_j = a_j / gA_j \]
\[ r_j = N_0_j / N_{Rj} \]
\[ \Delta x_j = L_j / N_{Rj} \]
\[ K_j = f \Delta x_j / 8 g D_j A_j^2 \]

% Junction analysis for each time step.
For \( i = 1 : N_j \) % Loop through junctions.
% Initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_t \]
\[ f = 0 \]

% Calculate dynamic constants for pipes adjacent to junction \( i \) [11 FLOPs/pipe].
For \( j \in \left[ N_{Bj}, N_{Aj} \right] \)
\[ Q_R = \begin{cases} Q_{(j,N_{Rj})} - r_j & j \in N_{Bj} \\ Q_{(j,2)} - r_j & j \in N_{Aj} \end{cases} \]
\[ H_R = \begin{cases} H_{(j,N_{Rj})} - r_j & j \in N_{Bj} \\ -H_{(j,2)} - r_j & j \in N_{Aj} \end{cases} \]
\[ C_6 = K_j \left| Q_{R} \right| \]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ C_{7j} = \varepsilon_j C_6 \quad (1) \]
\[ C_{8j} = C_6 - C_{7j} \quad (1) \]
\[ B_{Rj} = B_j + C_{7j} \quad (1) \]
\[ C_{10j} = 1/B_{Rj} \quad (1) \]
\[ B_0 = B_0 + C_{10j} \quad (1) \]
\[ C_{Rj} = (B_j - C_{8j}) Q_{Rj} + H_R \quad (3) \]
\[ C_{11j} = C_{Rj}/B_{Rj} \quad (1) \]
\[ C_0 = \begin{cases} C_0 + C_{11j} & j \in N_{B_i} \\ C_0 - C_{11j} & j \in N_{A_i} \end{cases} \quad (1) \]

If \( \theta_j > 0 \), \( f = 1 \)

% Calculate predictor head solution [1 FLOP].
\[ H_i = C_0/B_0 \quad (1) \]

% Calculate predictor flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ Q_{Pj} = \begin{cases} \left( C_{Rj} - H_i \right)/B_{Rj} & j \in N_{B_i} \\ \left( C_{Rj} + H_i \right)/B_{Rj} & j \in N_{A_i} \end{cases} \quad (2) \]

% Update dynamic constants for pipes adjacent to junction \( i \).
If \( f = 1 \)
% Re-initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]

% Update dynamic constants [11 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
If \( \theta_j > 0 \)
\[ Q_R = \begin{cases} Q_{(j,N_{Rj})}^{t-r_j} & j \in N_{B_i} \\ Q_{(j,2)}^{t-r_j} & j \in N_{A_i} \end{cases} \]
\[ B_{Rj} = B_{Rj} + \theta_j \left( K_j \left| Q_{Pj} \right| - C_{7j} \right) \quad (4) \]
\[ C_{10j} = 1/B_{Rj} \quad (1) \]
\[ C_{Rj} = C_{Rj} + \theta_j C_{8j} Q_{Rj} \quad (3) \]
\[ C_{11j} = C_{Rj}/B_{Rj} \quad (1) \]
\[ B_0 = B_0 + C_{10j} \quad (1) \]
\( C_0 = \begin{cases} C_0 + C_{11j} & j \in N_{B_i} \\ C_0 - C_{11j} & j \in N_{A_i} \end{cases} \) \hspace{1cm} (1)

% Calculate corrector head solution [1 FLOP].
\( H_i = C_0/B_0 \) \hspace{1cm} (1)

% Calculate corrector flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ Q_{P_j} = \begin{cases} \left( C_{R_j} - H_i \right) / B_{R_j} & j \in N_{B_i} \\ \left( C_{R_j} + H_i \right) / B_{R_j} & j \in N_{A_i} \end{cases} \] \hspace{1cm} (2)

### B4. GCM with Steady Friction and \( \varepsilon = 0 \)

#### B4.1 Compatibility Expressions and Solutions

Positive compatibility equation for interior and junction analyses:
\[ B_A Q_P + H_P - C_A = 0 \] \hspace{1cm} [B18]
\[ B_A = B + \theta K |Q_P| \]
\[ C_A = (B - (1 - \theta)K|Q_A|)Q_A + H_A \]

Negative compatibility equation for interior and junction analyses:
\[ B_B Q_P - H_P - C_B = 0 \] \hspace{1cm} [B19]
\[ B_B = B + \theta K |Q_P| \]
\[ C_B = (B - (1 - \theta)K|Q_B|)Q_B - H_B \]

Interior analysis solution:
\[ J^{(m)} Q_P^{(m+1)} = 2\theta K \Big| Q_P^{(m)} \Big| Q_P^{(m)} + C_A + C_B \] \hspace{1cm} [B20]
\[ J^{(m)} = 4\theta K \Big| Q_P^{(m)} \Big| + 2B \]
\[ H_P = C_A - B_A Q_P \] \hspace{1cm} [B21]

Junction analysis solution:
\[ H_P = \left( \sum_A \frac{1}{(B_A)_i} + \sum_B \frac{1}{(B_B)_i} \right)^{-1} \left( \sum_A \frac{C_A}{(B_A)_i} - \sum_B \frac{C_B}{(B_B)_i} - q_P \right) \] \hspace{1cm} [B22]
B4.2 Interior Analysis Implementation

% Pre-calculated static constants [12 FLOPs].

\[ B = \frac{a}{gA} \]  \hspace{1cm} (2)

\[ \Delta x_j = \frac{L_j}{N_{R_j}} \]  \hspace{1cm} (1)

\[ K = f\frac{\Delta x}{8gDA^2} \]  \hspace{1cm} (6)

\[ C_4 = 1 - \theta \]  \hspace{1cm} (1)

\[ C_7 = 2B \]  \hspace{1cm} (1)

\[ C_9 = 2\theta \]  \hspace{1cm} (1)

For \( j = 1: N_P \) % Loop through pipes.

For \( i = 2: N_R - 1 \) % Loop through interior nodes.

% Calculate dynamic constants [11 FLOPs/reach].

\[ C_{11} = K|Q_A| \]  \hspace{1cm} (1)

\[ C_{12} = K|Q_B| \]  \hspace{1cm} (1)

\[ C_A = (B - C_4C_{11})Q_A + H_A \]  \hspace{1cm} (4)

\[ C_B = (B - C_4C_{12})Q_B - H_B \]  \hspace{1cm} (4)

\[ C_{14} = C_A + C_B \]  \hspace{1cm} (1)

% Initialize flow estimate.

\[ Q_P^1 = Q_P^b \]  \hspace{1cm}

% Calculate flow solution [10 FLOPs/iteration-reach].

While true

\[ Q_P^0 = Q_P^1 \]  \hspace{1cm} (2)

\[ C_{15} = C_9K|Q_P^0| \]  \hspace{1cm} (1)

\[ C_{16} = 2C_{15} \]  \hspace{1cm} (1)

\[ J^m = C_{16} + C_7 \]  \hspace{1cm} (1)

\[ Q_P^1 = ((C_{16} - C_{15})Q_P^0 + C_{14})/J \]  \hspace{1cm} (4)

If \( \theta = 0 \) or \( |Q_P^1 - Q_P^0|/|Q_P^1| < e_{tol} \), break  \hspace{1cm} (2)

% Calculate head solution [5 FLOPs/reach].

\[ B_A = B + \theta K|Q_P^1| \]  \hspace{1cm} (3)

\[ H_P = C_A - B_AQ_P^b \]  \hspace{1cm} (2)

B4.3 Junction Analysis Implementation

% Calculate static constants [10 FLOPs/pipe].

\[ B_j = \frac{a_j}{gA_j} \]  \hspace{1cm} (2)

\[ r_j = \frac{N_{0_j}}{N_{R_j}} \]  \hspace{1cm} (1)

\[ \Delta x_j = \frac{L_j}{N_{R_j}} \]  \hspace{1cm} (1)
\[ K_j = f \Delta x_j / 8 g D_j A_j^2 \]  \hfill (6)

% Junction analysis for each time step.
For \( i = 1: N_j \)
% Initialize counters.
\( B_0 = 0 \)
\( C_0 = -q_i \)
\( f = 0 \)

% Calculate dynamic constants for pipes adjacent to junction \( i \) [8 FLOPs/pipe].
For \( j \in [N_{Bi}, N_{Ai}] \)
\[ Q_{Rj} = \begin{cases} 
 Q_{(j,NRj)}^{t-r_j} & j \in N_{Bi} \\
 Q_{(j,2)}^{t-r_j} & j \in N_{Ai} 
\end{cases} \]
\[ H_R = \begin{cases} 
 H_{(j,NRj)}^{t-r_j} & j \in N_{Bi} \\
 -H_{(j,2)}^{t-r_j} & j \in N_{Ai} 
\end{cases} \]
\[ C_{6j} = K_j \left| Q_{Rj} \right| \]  \hfill (1)
\[ B_{Rj} = B_j \]  \hfill (0)
\[ C_{10j} = 1 / B_{Rj} \]  \hfill (1)
\[ B_0 = B_0 + C_{10j} \]  \hfill (1)
\[ C_{Rj} = (B_j - C_{6j}) Q_{Rj} + H_R \]  \hfill (3)
\[ C_{11j} = C_{Rj} / B_{Rj} \]  \hfill (1)
\[ C_0 = \begin{cases} 
 C_0 + C_{11j} & j \in N_{Bi} \\
 C_0 - C_{11j} & j \in N_{Ai} 
\end{cases} \]  \hfill (1)
If \( \theta_j > 0 \), \( f = 1 \)

% Calculate predictor head solution [1 FLOP].
\[ H_i = C_0 / B_0 \]  \hfill (1)

% Calculate predictor flow solutions [2 FLOPs/pipe].
For \( j \in [N_{Bi}, N_{Ai}] \)
\[ Q_{Pj} = \begin{cases} 
 (C_{Rj} - H_i) / B_{Rj} & j \in N_{Bi} \\
 (C_{Rj} + H_i) / B_{Rj} & j \in N_{Ai} 
\end{cases} \]  \hfill (2)

% Update dynamic constants for pipes adjacent to junction \( i \).
If \( f = 1 \)

% Re-initialize counters.

\[ B_0 = 0 \]
\[ C_0 = -q_i \]

% Update dynamic constants [10 FLOPs/pipe].

For \( j \in [N_{B_i}, N_A] \)

If \( \theta_j > 0 \)

\[
Q_R = \begin{cases} 
Q_{(j,N_{R_j})}^{t-r_j} & j \in N_{B_i} \\
Q_{(j,2)}^{t-r_j} & j \in N_{A_i} 
\end{cases}
\]

\[ B_{R_j} = B_{R_j} + \theta_j K_j |Q_{P_j}| \]

\[ C_{10_j} = 1/B_{R_j} \]  \hspace{1cm} (1)

\[ C_{R_j} = C_{R_j} + \theta_j C_{R_j} Q_{R_j} \]

\[ C_{11_j} = C_{R_j}/B_{R_j} \]  \hspace{1cm} (1)

\[ B_0 = B_0 + C_{10_j} \]

\[ C_0 = \begin{cases} 
C_0 + C_{11_j} & j \in N_{B_i} \\
C_0 - C_{11_j} & j \in N_{A_i} 
\end{cases} \]  \hspace{1cm} (1)

% Calculate corrector head solution [1 FLOP].

\[ H_i = C_0/B_0 \]  \hspace{1cm} (1)

% Calculate corrector flow solutions [2 FLOPs/pipe].

For \( j \in [N_{B_i}, N_A] \)

\[
Q_{P_j} = \begin{cases} 
(C_{R_j} - H_i)/B_{R_j} & j \in N_{B_i} \\
(C_{R_j} + H_i)/B_{R_j} & j \in N_{A_i} 
\end{cases} \]  \hspace{1cm} (2)

**B5. GCM with Steady Friction, \( \theta = 0.5 \), and \( \varepsilon = 0 \)**

**B5.1 Compatibility Expressions and Solutions**

Positive compatibility equation for interior and junction analyses:

\[
B_A Q_P + H_P - C_A = 0 \quad \text{[B23]}
\]

\[
B_A = B + \frac{1}{2} K |Q_P| 
\]

\[
C_A = \left( B - \frac{1}{2} K |Q_A| \right) Q_A + H_A
\]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

Negative compatibility equation for interior and junction analyses:

\[ B_B Q_P - H_P - C_B = 0 \]  \hspace{1cm} [B24]

\[ B_B = B + \frac{1}{2} K |Q_P| \]

\[ C_B = \left( B - \frac{1}{2} K |Q_B| \right) Q_B - H_B \]

Interior analysis solution:

\[ J^{(m)} Q_P^{(m+1)} = K |Q_P^{(m)}| Q_P^{(m)} + C_A + C_B \]  \hspace{1cm} [B25]

\[ J^{(m)} = 2 \left( K |Q_P^{(m)}| + B \right) \]

\[ H_P = C_A - B_A Q_P \]  \hspace{1cm} [B26]

Junction analysis solution:

\[ H_P = \left( \sum_A \frac{1}{(B_A)_i} + \sum_B \frac{1}{(B_B)_i} \right)^{-1} \left( \sum_A \frac{(C_A)_i}{(B_A)_i} - \sum_B \frac{(C_B)_i}{(B_B)_i} - q_P \right) \]  \hspace{1cm} [B27]

B5.2 Interior Analysis Implementation

% Pre-calculated static constants [10 FLOPs].

\[ B = a / gA \quad (2) \]

\[ \Delta x_j = L_j / N_{R_j} \quad (1) \]

\[ K = f \Delta x / 8gDA^2 \quad (6) \]

\[ C_4 = K / 2 \quad (1) \]

For \( j = 1: N_P \) % Loop through pipes.

For \( i = 2: N_R - 1 \) % Loop through interior nodes.

% Calculate dynamic constants [9 FLOPs/reach].

\[ C_{11} = C_4 |Q_A| \quad (1) \]

\[ C_{12} = C_4 |Q_B| \quad (1) \]

\[ C_A = (B - C_{11}) Q_A + H_A \quad (3) \]

\[ C_B = (B - C_{12}) Q_B - H_B \quad (3) \]

\[ C_{14} = C_A + C_B \quad (1) \]

% Initialize flow estimate.

\[ Q_P^1 = Q_P^T \]

% Calculate flow solution [8 FLOPs/iteration-reach].

While true

\[ Q_P^0 = Q_P^1 \]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ C_{15} = K |Q_P^0| \]  \hspace{1cm} (1)
\[ f^m = 2(C_{15} + B) \]  \hspace{1cm} (2)
\[ Q_1^1 = (C_{15}Q_P^0 + C_{14})/J \]  \hspace{1cm} (3)
If \( \theta = 0 \) or \(|Q^1_P - Q_P^0|/|Q_P^0| < e_{tol} \), break

\% Calculate head solution [4 FLOPs/reach].
\[ B_A = B + C_4|Q_P^1| \]  \hspace{1cm} (2)
\[ H_P = C_A - B_A Q_P^1 \]  \hspace{1cm} (2)

**B5.3 Junction Analysis Implementation**

\% Calculate static constants [10 FLOPs/pipe].
\[ B_j = a_j / gA_j \]  \hspace{1cm} (2)
\[ r_j = N_{0j}/N_{Rj} \]  \hspace{1cm} (1)
\[ \Delta x_j = L_j/N_{Rj} \]  \hspace{1cm} (1)
\[ K_j = f\Delta x_j / 8gD_jA_j^2 \]  \hspace{1cm} (6)
% Junction analysis for each time step.
For \( i = 1: N_j \) % Loop through junctions.
\% Initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]
\[ f = 0 \]
% Calculate dynamic constants for pipes adjacent to junction \( i \) [8 FLOPs/pipe].
For \( j \in [N_{Bj}, N_{Aj}] \)
\[ Q_{Rj} = \begin{cases} 
Q_{(j,N_{Rj})}^{t-r_j} & j \in N_{Bj} \\
Q_{(j,2)}^{t-r_j} & j \in N_{Aj}
\end{cases} \]
\[ H_R = \begin{cases} 
H_{(j,N_{Rj})}^{t-r_j} & j \in N_{Bj} \\
-H_{(j,2)}^{t-r_j} & j \in N_{Aj}
\end{cases} \]
\[ C_{6j} = K_j |Q_{Rj}| \]  \hspace{1cm} (1)
\[ B_{Rj} = B_j \]  \hspace{1cm} (0)
\[ C_{10j} = 1/B_{Rj} \]  \hspace{1cm} (1)
\[ B_0 = B_0 + C_{10j} \]  \hspace{1cm} (1)
\[ C_{Rj} = (B_j - C_{6j})Q_{Rj} + H_R \]  \hspace{1cm} (3)
\[ C_{11j} = C_{Rj}/B_{Rj} \]  \hspace{1cm} (1)
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ C_0 = \begin{cases} C_0 + C_{11j} & j \in N_{B_i} \\ C_0 - C_{11j} & j \in N_{A_i} \end{cases} \]  
(1)

\[ \text{If } \theta_j > 0, f = 1 \]

% Calculate predictor head solution [1 FLOP].
\[ H_i = C_0 / B_0 \]  
(1)

% Calculate predictor flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ Q_{Pj} = \begin{cases} \left( C_{Rj} - H_i \right) / B_{Rj} & j \in N_{B_i} \\ \left( C_{Rj} + H_i \right) / B_{Rj} & j \in N_{A_i} \end{cases} \]  
(2)

% Update dynamic constants for pipes adjacent to junction \( i \).
If \( f = 1 \)
\[ \text{If } f = 1 \]
% Re-initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]

% Update dynamic constants [10 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ Q_{R} = \begin{cases} Q_{(j,N_{Rj})} & j \in N_{B_i} \\ Q_{(j,2)} & j \in N_{A_i} \end{cases} \]
\[ B_{Rj} = B_{Rj} + K_j \left| Q_{Pj} \right| / 2 \]  
(3)
\[ C_{10j} = 1 / B_{Rj} \]  
(1)
\[ C_{Rj} = C_{Rj} + C_{6j} Q_{Rj} / 2 \]  
(3)
\[ C_{11j} = C_{Rj} / B_{Rj} \]  
(1)
\[ B_0 = B_0 + C_{10j} \]  
(1)
\[ C_0 = \begin{cases} C_0 + C_{11j} & j \in N_{B_i} \\ C_0 - C_{11j} & j \in N_{A_i} \end{cases} \]  
(1)

% Calculate corrector head solution [1 FLOP].
\[ H_i = C_0 / B_0 \]  
(1)

% Calculate corrector flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ Q_{P_j} = \begin{cases} 
\left( C_{R_j} - H_i \right) / B_{R_j} & j \in N_{B_i} \\
\left( C_{R_j} + H_i \right) / B_{R_j} & j \in N_{A_i}
\end{cases} \] (2)

B6. GCM with Steady Friction, \( \theta = 1 \), and \( \varepsilon = 0 \)

B6.1 Compatibility Expressions and Solutions

Positive compatibility equation for interior and junction analyses:

\[ B_A Q_P + H_P - C_A = 0 \] [B28]
\[ B_A = B + K|Q_P| \]
\[ C_A = BQ_A + H_A \]

Negative compatibility equation for interior and junction analyses:

\[ B_B Q_P - H_P - C_B = 0 \] [B29]
\[ B_B = B + K|Q_P| \]
\[ C_B = BQ_B - H_B \]

Interior analysis solution:

\[ J^{(m)} Q_P^{(m+1)} = 2K \left| Q_P^{(m)} \right| Q_P^{(m)} + C_A + C_B \] [B30]
\[ J^{(m)} = 4K \left| Q_P^{(m)} \right| + 2B \]
\[ H_P = C_A - B_A Q_P \] [B31]

Junction analysis solution:

\[ H_P = \left( \sum_A \frac{1}{(B_A)_i} + \sum_B \frac{1}{(B_B)_i} \right)^{-1} \left( \sum_A \frac{(C_A)_i}{(B_A)_i} - \sum_B \frac{(C_B)_i}{(B_B)_i} - q_P \right) \] [B32]

B6.2 Interior Analysis Implementation

% Pre-calculated static constants [10 FLOPs].
\[ B = a/gA \] (2)
\[ \Delta x_j = L_j/N_{R_j} \] (1)
\[ K = f \Delta x/8gDA^2 \] (6)
\[ C_7 = 2B \] (1)

For \( j = 1 \): \( N_P \) % Loop through pipes.
For \( i = 2 \): \( N_R - 1 \) % Loop through interior nodes.
Appendix B: Generalized Characteristic Method
Implementation Algorithms

% Calculate dynamic constants [7 FLOPs/reach].
\[ C_{11} = K|Q_A| \quad (1) \]
\[ C_{12} = K|Q_B| \quad (1) \]
\[ C_A = BQ_A + H_A \quad (2) \]
\[ C_B = BQ_B - H_B \quad (2) \]
\[ C_{14} = C_A + C_B \quad (1) \]

% Initialize flow estimate.
\[ Q_{P_1}^1 = Q_{P_0}^0 \]

% Calculate flow solution [10 FLOPs/iteration-reach].
While true
\[ Q_{P_0}^0 = Q_{P_1}^1 \]
\[ C_{15} = 2K|Q_{P_0}^0| \quad (2) \]
\[ C_{16} = 2C_{15} \quad (1) \]
\[ J^m = C_{16} + C_7 \quad (1) \]
\[ Q_{P_1}^1 = \frac{((C_{16} - C_{15})Q_{P_0}^0 + C_{14})}{J} \quad (4) \]
If \( \theta = 0 \) or \[ |Q_{P_1}^1 - Q_{P_0}^0|/|Q_{P_1}^1| < e_{tot} \], break

% Calculate head solution [4 FLOPs/reach].
\[ B_A = B + K|Q_{P_1}^1| \quad (2) \]
\[ H_P = C_A - B_AQ_{P_1}^1 \quad (2) \]

B6.3 Junction Analysis Implementation

% Calculate static constants [10 FLOPs/pipe].
\[ B_j = a_j/gA_j \quad (2) \]
\[ r_j = N_{0j}/N_{Rj} \quad (1) \]
\[ \Delta x_j = L_j/N_{Rj} \quad (1) \]
\[ K_j = f\Delta x_j/8gD_jA_j^2 \quad (6) \]

% Junction analysis for each time step.
For \( i = 1: N_j \) % Loop through junctions.
% Initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]
\[ f = 0 \]

% Calculate dynamic constants for pipes adjacent to junction \( i \) [8 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ Q_{Rj} = \begin{cases} Q_{(j, NRj)}^{t-r_j} & j \in N_{Bi} \\ Q_{(j, 2)}^{t-r_j} & j \in N_{Ai} \end{cases} \]

\[ H_R = \begin{cases} H_{(j, NRj)}^{t-r_j} & j \in N_{Bi} \\ -H_{(j, 2)}^{t-r_j} & j \in N_{Ai} \end{cases} \]

\[ C_{6j} = K_j \left| Q_{Rj} \right| \]

\[ B_{Rj} = B_j \]

\[ C_{10j} = 1/B_{Rj} \]

\[ B_0 = B_0 + C_{10j} \]

\[ C_{Rj} = (B_j - C_{6j}) Q_{Rj} + H_R \]

\[ C_{11j} = C_{Rj}/B_{Rj} \]

\[ C_0 = \begin{cases} C_0 + C_{11j} & j \in N_{Bi} \\ C_0 - C_{11j} & j \in N_{Ai} \end{cases} \]

If \( \theta_j > 0 \), \( f = 1 \)

% Calculate predictor head solution [1 FLOP].
\[ H_i = C_0/B_0 \]

% Calculate predictor flow solutions [2 FLOPs/pipe].
For \( j \in [N_{Bi}, N_{Ai}] \)
\[ Q_{Pj} = \begin{cases} \left( C_{Rj} - H_i \right)/B_{Rj} & j \in N_{Bi} \\ \left( C_{Rj} + H_i \right)/B_{Rj} & j \in N_{Ai} \end{cases} \]

% Update dynamic constants for pipes adjacent to junction \( i \).
If \( f = 1 \)
% Re-initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]

% Update dynamic constants [8 FLOPs/pipe].
For \( j \in [N_{Bi}, N_{Ai}] \)
If \( \theta_j > 0 \)
\[ Q_R = \begin{cases} Q_{(j, NRj)}^{t-r_j} & j \in N_{Bi} \\ Q_{(j, 2)}^{t-r_j} & j \in N_{Ai} \end{cases} \]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ B_R j = B_R j + K_j \left| Q_{Pj} \right| \]  \hfill (2)
\[ C_{10j} = 1/B_{Rj} \]  \hfill (1)
\[ C_{Rj} = C_{Rj} + C_{6j} Q_{Rj} \]  \hfill (2)
\[ C_{11j} = C_{Rj}/B_{Rj} \]  \hfill (1)
\[ B_0 = B_0 + C_{10j} \]  \hfill (1)
\[ C_0 = \begin{cases} C_0 + C_{11j} & j \in N_{Bi} \\ C_0 - C_{11j} & j \in N_{Ai} \end{cases} \]  \hfill (1)

% Calculate corrector head solution [1 FLOP].
\[ H_i = C_0/B_0 \]  \hfill (1)

% Calculate corrector flow solutions [2 FLOPs/pipe].
For \( j \in [N_{Bi}, N_{Ai}] \)
\[ Q_{Pj} = \begin{cases} (C_{Rj} - H_i)/B_{Rj} & j \in N_{Bi} \\ (C_{Rj} + H_i)/B_{Rj} & j \in N_{Ai} \end{cases} \]  \hfill (2)

B7. GCM with Steady Friction and \( \theta = 0 \)

B7.1 Compatibility Expressions and Solutions

Positive compatibility equation for interior and junction analyses:
\[ B_A Q_P + H_P - C_A = 0 \]  \hfill [B33]
\[ B_A = B + \varepsilon K |Q_A| \]
\[ C_A = (B - (1 - \varepsilon)K|Q_A|)Q_A + H_A \]

Negative compatibility equation for interior and junction analyses:
\[ B_B Q_P - H_P - C_B = 0 \]  \hfill [B34]
\[ B_B = B + \varepsilon K |Q_B| \]
\[ C_B = (B - (1 - \varepsilon)K|Q_B|)Q_B - H_B \]

Interior analysis solution:
\[ Q_P = (C_A + C_B)/J \]  \hfill [B35]
\[ J = 2B + \varepsilon K(|Q_A| + |Q_B|) \]
\[ H_P = C_A - B_A Q_P \]  \hfill [B36]
Junction analysis solution:

\[
H_P = \left( \sum_A \frac{1}{[B_A]} + \sum_B \frac{1}{[B_B]} \right)^{-1} \left( \sum_A \frac{[C_A]}{[B_A]} - \sum_B \frac{[C_B]}{[B_B]} - q_P \right) \tag{B37}
\]

### B7.2 Interior Analysis Implementation

% Pre-calculated static constants [10 FLOPs].

\[
B = \frac{a}{gA} \tag{2}
\]

\[
\Delta x_j = \frac{L_j}{N_{Rj}} \tag{1}
\]

\[
K = f \Delta x / 8gDA^2 \tag{6}
\]

\[
C_5 = 1 - \varepsilon \tag{1}
\]

For \( j = 1: N_P \) % Loop through pipes.

For \( i = 2: N_R - 1 \) % Loop through interior nodes.

% Calculate dynamic constants [14 FLOPs/reach].

\[
C_{11} = K|Q_A| \tag{1}
\]

\[
C_{12} = K|Q_B| \tag{1}
\]

\[
C_A = (B - C_5 C_{11})Q_A + H_A \tag{4}
\]

\[
C_B = (B - C_5 C_{12})Q_B - H_B \tag{4}
\]

\[
C_{13} = 2B + \varepsilon (C_{11} + C_{12}) \tag{4}
\]

% Calculate flow and head solutions [6 FLOPs/reach].

\[
Q_P = \frac{(C_A + C_B)}{C_{13}} \tag{2}
\]

\[
B_A = B + \varepsilon C_{11} \tag{2}
\]

\[
H_P = C_A - B_A Q_P \tag{2}
\]

### B7.3 Junction Analysis Implementation

% Calculate static constants [10 FLOPs/pipe].

\[
B_j = \frac{a_j}{gA_j} \tag{2}
\]

\[
r_j = \frac{N_{0j}}{N_{Rj}} \tag{1}
\]

\[
\Delta x_j = \frac{L_j}{N_{Rj}} \tag{1}
\]

\[
K_j = f \Delta x_j / 8gD_j A_j^2 \tag{6}
\]

% Junction analysis for each time step.

For \( i = 1: N_j \) % Loop through junctions.

% Initialize counters.

\[
B_0 = 0
\]

\[
C_0 = -q_i
\]

\[
f = 0
\]
% Calculate dynamic constants for pipes adjacent to junction $i$ [11 FLOPs/pipe].

For $j \in [N_{Bi}, N_{Ai}]$

\[
Q_{Rj} = \begin{cases} 
Q_{(i,N_{Rj})}^{t-r_{j}} & j \in N_{Bi} \\
Q_{(j,2)}^{t-r_{j}} & j \in N_{Ai}
\end{cases}
\]

\[
H_{R} = \begin{cases} 
H_{(j,N_{Rj})}^{t-r_{j}} & j \in N_{Bi} \\
-H_{(j,2)}^{t-r_{j}} & j \in N_{Ai}
\end{cases}
\]

\[C_{6} = K_{j} \left| Q_{Rj} \right| \quad (1)\]

\[C_{7} = \varepsilon_{j} C_{6} \quad (1)\]

\[B_{Rj} = B_{j} + C_{7} \quad (1)\]

\[B_{0} = B_{0} + 1/B_{Rj} \quad (2)\]

\[C_{Rj} = \left( B_{j} - C_{6} + C_{7} \right) Q_{Rj} + H_{R} \quad (4)\]

\[C_{0} = \begin{cases} 
C_{0} + C_{Rj}/B_{Rj} & j \in N_{Bi} \\
C_{0} - C_{Rj}/B_{Rj} & j \in N_{Ai}
\end{cases} \quad (2)\]

% Calculate head solution [1 FLOP].

\[H_{i} = C_{0}/B_{0} \quad (1)\]

% Calculate flow solutions [2 FLOPs/pipe].

For $j \in [N_{Bi}, N_{Ai}]$

\[
Q_{Pj} = \begin{cases} 
\left( C_{Rj} - H_{i} \right) / B_{Rj} & j \in N_{Bi} \\
\left( C_{Rj} + H_{i} \right) / B_{Rj} & j \in N_{Ai}
\end{cases} \quad (2)
\]

**B8. GCM with Steady Friction, $\theta = 0$, and $\varepsilon = 0.5$**

**B8.1 Compatibility Expressions and Solutions**

Positive compatibility equation for interior and junction analyses:

\[B_{A} Q_{P} + H_{P} - C_{A} = 0 \quad [B38]\]

\[B_{A} = B + \frac{1}{2} K |Q_{A}|\]

\[C_{A} = \left( B - \frac{1}{2} K |Q_{A}| \right) Q_{A} + H_{A}\]

Negative compatibility equation for interior and junction analyses:
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ B_B Q_P - H_P - C_B = 0 \]  \[ B_B = B + \frac{1}{2} K|Q_B| \]  \[ C_B = \left( B - \frac{1}{2} K|Q_B| \right) Q_B - H_B \]

Interior analysis solution:

\[ Q_P = (C_A + C_B)/J \]  \[ J = 2B + \frac{1}{2} K(|Q_A| + |Q_B|) \]

\[ H_P = C_A - B_A Q_P \]

Junction analysis solution:

\[ H_P = \left( \sum_A \frac{1}{(B_A)_i} + \sum_B \frac{1}{(B_B)_i} \right)^{-1} \left( \sum_A \frac{(C_A)_i}{(B_A)_i} - \sum_B \frac{(C_B)_i}{(B_B)_i} - q_P \right) \]

### B8.2 Interior Analysis Implementation

% Pre-calculated static constants [9 FLOPs].

\[ B = a/gA \]  \[ \Delta x_j = L_j/N_{R_j} \]  \[ K = f\Delta x/8gDA^2 \]

For \( j = 1: N_P \) % Loop through pipes.

For \( i = 2: N_R - 1 \) % Loop through interior nodes.

% Calculate dynamic constants [13 FLOPs/reach].

\[ C_{11} = K|Q_A| \]  \[ C_{11b} = C_{11}/2 \]  \[ C_{12} = K|Q_B|/2 \]

\[ C_A = (B - C_{11b})Q_A + H_A \]  \[ C_B = (B - C_{12})Q_B - H_B \]  \[ C_{13} = 2B + C_{11b} + C_{12} \]

% Calculate flow and head solutions [5 FLOPs/reach].

\[ Q_P = (C_A + C_B)/C_{13} \]  \[ B_A = B + C_{11b} \]  \[ H_P = C_A - B_A Q_P \]

### B8.3 Junction Analysis Implementation

% Calculate static constants [10 FLOPs/pipe].
Appendix B: Generalized Characteristic Method  
Implementation Algorithms

\[ B_j = a_j \frac{1}{gA_j} \quad (2) \]
\[ r_j = N_{0j}/N_{Rj} \quad (1) \]
\[ \Delta x_j = L_j/N_{Rj} \quad (1) \]
\[ K_j = f \frac{\Delta x_j}{8 gD_jA_j^2} \quad (6) \]

% Junction analysis for each time step.
For \( i = 1 : N_j \) % Loop through junctions.
% Initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]
\[ f = 0 \]

% Calculate dynamic constants for pipes adjacent to junction \( i \) [10 FLOPs/pipe].
For \( j \in [N_{Bj}, N_{Aj}] \)
\[
Q_{Rj} = \begin{cases} 
Q_{(j,N_{Rj})} & j \in N_{Bj} \\
Q_{(j,2)} & j \in N_{Aj} 
\end{cases} 
\]
\[
H_R = \begin{cases} 
H_{(j,N_{Rj})} & j \in N_{Bj} \\
-H_{(j,2)} & j \in N_{Aj} 
\end{cases} 
\]
\[ C_7 = K_j \left| Q_{Rj} \right| /2 \quad (2) \]
\[ B_{Rj} = B_j + C_7 \quad (1) \]
\[ B_0 = B_0 + 1/B_{Rj} \quad (2) \]
\[ C_{Rj} = (B_j - C_7)Q_{Rj} + H_R \quad (3) \]
\[ C_0 = \begin{cases} 
C_0 + C_{Rj}/B_{Rj} & j \in N_{Bj} \\
C_0 - C_{Rj}/B_{Rj} & j \in N_{Aj} 
\end{cases} \quad (2) \]

% Calculate head solution [1 FLOP].
\[ H_i = C_0/B_0 \quad (1) \]

% Calculate flow solutions [2 FLOPs/pipe].
For \( j \in [N_{Bj}, N_{Aj}] \)
\[
Q_{Pj} = \begin{cases} 
\left( C_{Rj} - H_i \right)/B_{Rj} & j \in N_{Bj} \\
\left( C_{Rj} + H_i \right)/B_{Rj} & j \in N_{Aj} 
\end{cases} \quad (2) \]
B9. GCM with Steady Friction, $\theta = 0$, and $\varepsilon = 1$

B9.1 Compatibility Expressions and Solutions

Positive compatibility equation for interior and junction analyses:

$$B_A Q_P + H_P - C_A = 0 \quad \text{[B43]}$$

$$B_A = B + K|Q_A|$$

$$C_A = BQ_A + H_A$$

Negative compatibility equation for interior and junction analyses:

$$B_B Q_P - H_P - C_B = 0 \quad \text{[B44]}$$

$$B_B = B + K|Q_B|$$

$$C_B = BQ_B - H_B$$

Interior analysis solution:

$$Q_P = (C_A + C_B)/J \quad \text{[B45]}$$

$$J = 2B + K(|Q_A| + |Q_B|)$$

$$H_P = C_A - B_A Q_P \quad \text{[B46]}$$

Junction analysis solution:

$$H_P = \left(\sum_A \frac{1}{(B_A)_i} + \sum_B \frac{1}{(B_B)_i}\right)^{-1} \left(\sum_A \frac{(C_A)_i}{(B_A)_i} - \sum_B \frac{(C_B)_i}{(B_B)_i} - q_P\right) \quad \text{[B47]}$$

B9.2 Interior Analysis Implementation

% Pre-calculated static constants [9 FLOPs].

$$B = a/gA \quad \text{(2)}$$

$$\Delta x_j = L_j/N_{R_j} \quad \text{(1)}$$

$$K = f\Delta x/8gDA^2 \quad \text{(6)}$$

For $j = 1$: $N_p$ % Loop through pipes.

For $i = 2$: $N_{R} - 1$ % Loop through interior nodes.

% Calculate dynamic constants [9 FLOPs/each].

$$C_{11} = K|Q_A| \quad \text{(1)}$$

$$C_A = BQ_A + H_A \quad \text{(2)}$$

$$C_B = BQ_B - H_B \quad \text{(2)}$$

$$C_{13} = 2B + C_{11} + K|Q_B| \quad \text{(4)}$$
Appendix B: Generalized Characteristic Method
Implementation Algorithms

% Calculate flow and head solutions [5 FLOPs/reach].
\[
Q_P = (C_A + C_B)/C_{13} \\
B_A = B + C_{11} \\
H_P = C_A - B_A Q_P
\]  

(2)  

(1)  

(2)

B9.3 Junction Analysis Implementation

% Calculate static constants [10 FLOPs/pipe].
\[
B_j = a_j/gA_j \\
r_j = N_{0j}/N_{Rj} \\
\Delta x_j = L_j/N_{Rj} \\
K_j = f\Delta x_j/8gD_jA_j^2
\]  

(2)  

(1)  

(1)  

(6)

% Junction analysis for each time step.
For \( i = 1: N_j \) % Loop through junctions.
% Initialize counters.
\[
B_0 = 0 \\
C_0 = -q_i \\
f = 0
\]

% Calculate dynamic constants for pipes adjacent to junction \( i \) [8 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[
Q_{Rj} = \begin{cases} 
Q_{(j,N_{Rj})}^{t-r_j} & j \in N_{B_i} \\
Q_{(j,2)}^{t-r_j} & j \in N_{A_i}
\end{cases} \\
H_R = \begin{cases} 
H_{(j,N_{Rj})}^{t-r_j} & j \in N_{B_i} \\
-H_{(j,2)}^{t-r_j} & j \in N_{A_i}
\end{cases} \\
C_6 = K_j \left| Q_{Rj} \right| \\
B_{Rj} = B_j + C_6 \\
B_0 = B_0 + 1/B_{Rj} \\
C_{Rj} = B_j Q_{Rj} + H_R \\
C_0 = \begin{cases} 
C_0 + C_{Rj}/B_{Rj} & j \in N_{B_i} \\
C_0 - C_{Rj}/B_{Rj} & j \in N_{A_i}
\end{cases}
\]  

(1)  

(1)  

(2)  

(2)  

(2)

% Calculate head solution [1 FLOP].
\[
H_i = C_0/B_0
\]  

(1)
% Calculate flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[
Q_{P_j} = \begin{cases} 
(C_{R_j} - H_i)/B_{R_j} & j \in N_{B_i} \\
(C_{R_j} + H_i)/B_{R_j} & j \in N_{A_i}
\end{cases}
\]

(2)

**B10. GCM with Steady Friction and \( \theta = \varepsilon = 0 \)**

**B10.1 Compatibility Expressions and Solutions**

Positive compatibility equation for interior and junction analyses:
\[
B Q_P + H_P - C_A = 0 
\]
\[\text{[B48]}\]
\[
C_A = (B - K|Q_A|)Q_A + H_A
\]

Negative compatibility equation for interior and junction analyses:
\[
B Q_P - H_P - C_B = 0 
\]
\[\text{[B49]}\]
\[
C_B = (B - K|Q_B|)Q_B - H_B
\]

Interior analysis solution:
\[
Q_P = (C_A + C_B)/2B 
\]
\[\text{[B50]}\]
\[
H_P = C_A - B Q_P 
\]
\[\text{[B51]}\]

Junction analysis solution:
\[
H_P = \left( \sum_A \frac{1}{[B_A]_i} + \sum_B \frac{1}{[B_B]_i} \right)^{-1} \left( \sum_A \frac{[C_A]_i}{[B_A]_i} - \sum_B \frac{[C_B]_i}{[B_B]_i} - q_P \right) 
\]
\[\text{[B52]}\]

**B10.2 Interior Analysis Implementation**

% Pre-calculated static constants [9 FLOPs].
\[
B = a/gA 
\]
\[\text{(2)}\]
\[
\Delta x_j = L_j/N_{R_j} 
\]
\[\text{(1)}\]
\[
K = f\Delta x/8gDA^2 
\]
\[\text{(6)}\]

For \( j = 1: N_P \) % Loop through pipes.
For \( i = 2: N_R - 1 \) % Loop through interior nodes.
% Calculate dynamic constants [8 FLOPs/reach].
\[
C_{11} = K|Q_A| 
\]
\[\text{(1)}\]
Appendix B: Generalized Characteristic Method
Implementation Algorithms

\[ C_A = (B - C_{11}) Q_A + H_A \]  \hspace{1cm} (3)

\[ C_B = (B - K|Q_B|) Q_B - H_B \]  \hspace{1cm} (4)

\% Calculate flow and head solutions [5 FLOPs/reach].
\[ Q_P = (C_A + C_B)/2B \]  \hspace{1cm} (3)
\[ H_P = C_A - BQ_P \]  \hspace{1cm} (2)

B10.3 Junction Analysis Implementation

\% Calculate static constants [10 FLOPs/pipe].
\[ B_j = \alpha_j/gA_j \]  \hspace{1cm} (2)
\[ r_j = N_0_j/N_{R_j} \]  \hspace{1cm} (1)
\[ \Delta x_j = L_j/N_{R_j} \]  \hspace{1cm} (1)
\[ K_j = f\Delta x_j/8gD_jA_j^2 \]  \hspace{1cm} (6)

\% Junction analysis for each time step.
For \( i = 1: N_j \) \% Loop through junctions.
\% Initialize counters.
\[ B_0 = 0 \]
\[ C_0 = -q_i \]
\[ f = 0 \]

\% Calculate dynamic constants for pipes adjacent to junction \( i \) [8 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[ Q_{R_j} = \begin{cases} Q_{(j,N_{R_j})}^{t-r_j} & j \in N_{B_i} \\ Q_{(j,2)}^{t-r_j} & j \in N_{A_i} \end{cases} \]
\[ H_R = \begin{cases} H_{(j,N_{R_j})}^{t-r_j} & j \in N_{B_i} \\ -H_{(j,2)}^{t-r_j} & j \in N_{A_i} \end{cases} \]
\[ C_6 = K_j \left| Q_{R_j} \right| \]  \hspace{1cm} (1)
\[ B_0 = B_0 + 1/B_j \]  \hspace{1cm} (2)
\[ C_{R_j} = (B_j - C_6) Q_{R_j} + H_R \]  \hspace{1cm} (3)
\[ C_0 = \begin{cases} C_0 + C_{R_j}/B_j & j \in N_{B_i} \\ C_0 - C_{R_j}/B_j & j \in N_{A_i} \end{cases} \]  \hspace{1cm} (2)

\% Calculate head solution [1 FLOP].
\[ H_i = C_0/B_0 \]  \hspace{1cm} (1)
% Calculate flow solutions [2 FLOPs/pipe].
For \( j \in [N_{B_i}, N_{A_i}] \)
\[
Q_{Pj} = \begin{cases} 
\left( \frac{C_{Rj} - H_i}{B_j} \right) & j \in N_{B_i} \\
\left( \frac{C_{Rj} + H_i}{B_j} \right) & j \in N_{A_i}
\end{cases}
\] (2)