PID CONTROLLED ADAPTIVE TIME-STEPPING IN COUPLED SURFACE-SUBSURFACE SIMULATION: A TOOL FOR REDUCING NON-PHYSICAL OSCILLATION

A Thesis

by

BRIAN SAMUEL REDICK

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Eduardo Gildin
Co-Chair of Committee,	Jihoon Kim
Committee Member,	Shankar P. Bhattacharyya
Head of Department,	A. Daniel Hill

August 2017

Major Subject: Petroleum Engineering

ABSTRACT

Surface-subsurface coupling in simulation is required to model large, complex and often offshore projects. The most optimal form of coupling is the partially implicit method. The partially implicit method typically balances accuracy and computational costs. However, the partially implicit method can suffer from non-physical oscillations. Non-physical oscillations are a result of incorrect assumptions made during coupled simulation. Non-physical oscillations are solely artifacts of less than ideal coupled simulation- they do not have physical significance. As such, non-physical oscillations in coupling are treated as simulation complications, as opposed to reservoir or network dynamics. Although many coupling solutions exist, many are complex and difficult to implement in commercial software packages. In this study, we investigate the use of PID control to select time-steps in coupled surface-subsurface simulation. PID control is not often used in oil and gas applications, however it is well established in other engineering fields. The PID controller performs automatic, adaptive time-stepping in the coupled simulation. The controller operated by reducing oscillations in coupling error. The results show that the PID controlled time-stepping reduces non-physical oscillations, the total number of time-steps executed, and the computational cost of coupled simulation. Importantly, one PID controlled experiment decreased simulation time by 300%, with less than 0.5% error (in cumulative production) as compared to Schlumberger's standard coupling settings. We performed a set of manual tuning experiments that highlight opportunity for controller optimization, as well as motivate future work. The PID controlled coupled simulation we created can be implemented in any software where the user can select the time-steps.

DEDICATION

This work is primarily dedicated to Kenneth Kubos, PhD. Dr. Kubos has positively impacted my life in many ways. Dr. Kubos is the best uncle anyone could ask for. Dr. Kubos is also my greatest mentor, my best friend, and constant confidant. He supported me through my undergraduate and graduate degrees, and supplied encouragement and moral support the whole way through. I am eternally grateful for Dr. Kubos's love, friendship and support. I can only strive to be like him.

This study would not be possible without my great friends at Texas A&M. I would like to thank Enrique Zárate Losoya for his constant friendship in the office, as well as for lending me his great intellect on a daily basis. I would also like to thank all of my friends in Dr. Gildin's Group: Alex Tarankanov, Jaewook Lee, Rafael Wanderley de Holanda, Tangmo Suranetinai, Marcelo Dall'Aqua, Pattanapong Plukmonton, Hardik Zalavadia, and Nadav Sorek. I learned many lessons from my friends in the office.

I would also like to mention my dear friends from the IFP: François "Coquin Bernache" Jumeau, Guillaume "Gillarm" Lostis, Romain Lemoine, Olivier "LC" Lançone, Celine Liron, and my friends from down under, Josiah "Spiders" Khor and Aaron Mendonsa. In addition, my American friends Ali "Billy Joe" Dhukka, Bartosz "Barry M" Czernia, Joshua Hernandez, Zach Benamram, and Jason Jackson provided great friendship and comradery. My colleagues from SPE also provided great friendship. Thank you Riyan Ariwibowo, Jordan Argamany, and Miranda Jones.

I would also like to mention my best childhood friend, Ryan Koch. Our paths have always been somehow parallel.

I would also like to thank Rohr Chabad of College Station for providing an unparalleled community that became home. I have much gratitude for Manya and Yossi Lazaroff, as well as my friends Justin Farrell, Elad Dermer, Jonathan Ben-Benjamin, and all others I spent many memorable Fridays with.

ACKNOWLEDGEMENTS

I would like to acknowledge Dr. Gildin for being a great mentor during my time at Texas A&M. I learned many lessons under his watch than I would have otherwise. I sincerely enjoyed my time as one of his students. I would also like to thank my committee members Dr. Kim and Dr. Bhattacharyya. I would also like to thank Dr. Barrufet for attending my defense in lieu of Dr. Kim.

I would also like to acknowledge Dr. Thomas Blasingame and Ms. Betty Robbins. Both were excellent advisors within Texas A&M SPE, and I immensely enjoyed working and learning from them both. I would like to acknowledge their great voluntary efforts within SPE. Their only reward is the professional and personal development of the students they advise, and of course, all student's admiration and respect.

I would like to acknowledge Vanguard Natural Resources, LLC (NASDAQ: VNR) for providing great internship opportunities across multiple summers and winters. These experiences provided excellent learning and work opportunities, as well as provided me a means to afford graduate education. I would also like to thank Keith Froebel for introducing me to Texas A&M. My time at Texas A&M has been the most memorable, challenging and rewarding two years of my life.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Professor Gildin [advisor], Professor Kim [co-chair] of the Harold Vance Department of Petroleum Engineering, as well as Professor Bhattacharyya [member] of the Department of Electrical and Computer Engineering.

All work for the thesis was completed independently by the student. There was no outside funding that supported this research. The student funded his own research.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION AND LITERATURE REVIEW

Reservoir and surface network coupling is a well-defined problem, and is especially pertinent to offshore asset design. Typical offshore production configurations feature a small number of highly productive wells producing to centralized production facilities and surface networks. In such cases, production facilities and other surface networks force operating constraints on reservoirs. Production rates, cumulative production and other metrics of reservoir performance are impacted by the configuration and operating conditions of surface networks. Therefore, when making business decisions, it is critical to couple both surface and subsurface systems in simulation. Surface-subsurface coupling allows engineers to accurately forecast production and facilitate the optimization of the reservoir, well, network and ultimately, the field development plan.

Engineers have many ways to couple simulators. Coupling methods fall into three categories: explicit coupling, partially explicit coupling, and fully implicit coupling. Explicit coupling is the loosest, or lowest fidelity coupling methodology. In explicit coupling, a set of separate simulators (e.g. a standalone reservoir simulator package and a standalone network multi-phase flow simulator package) are utilized. The simulators both use bottom hole pressure, tubing head pressure and well flow rates in their respective calculations. Both simulators are initialized with the same boundary conditions, such as equivalent bottom hole pressures. During simulation, the independent simulations are solved sequentially at equivalent time-steps. At the beginning of each time-step, the network simulator will typically pass its earlier (previous time-step) calculated solution for the bottom hole pressure to the reservoir simulator. The reservoir simulator assimilates this value, and uses as a boundary condition for the duration of the time-step, $t+\Delta t$. The reservoir simulator effectively calculates new pressures in the gridblocks. However, well models in reservoir simulation are based upon steady state radial flow, i.e. the Peaceman Equation. Therefore, the IPR calculations assume that the pressure in the completed gridblock remains constant across $t+\Delta t$: this is the so-called steady-state IPR. For short periods of transient pressure behavior in the reservoir, the steady-state simulated IPR does not accurately capture well performance. That is, the simulation assumes a steady-state IPR, when in fact the completed gridblock pressure changes significantly. The IPR is in effect overestimated during the simulation, and results in pressures and rates that exceed real, transient production. During the next time-step, the reservoir simulator responds to the earlier, large IPR by drastically reducing the pressures in the completed gridblock and surrounding gridblocks. The new, reduced IPR curve yields a drastically smaller production rate and pressure. This cycle repeats itself during periods of pressure transience. These cycles are referred to non-physical oscillations in coupling terminology. For steady-state pressure conditions in the reservoir, coupling is not an issue. However, as explained, explicitly coupled simulations will have non-physical production and pressure profiles during periods of transient pressure behavior in the reservoir. In addition, boundary conditions (e.g., bottom hole pressure) calculated by both simulators diverge across time-steps, as there are no operations taken to enforce convergence. Thus, the explicit method is almost always unstable.

The second category of coupling, partially implicit coupling, features various heuristics and modified calculations to overcome the issue of non-physical oscillation in explicit systems. Like explicitly coupled systems, partially implicit coupled systems utilize separate surface and subsurface simulators. However, partially implicit coupled simulations employ algorithms to promote boundary condition agreement between the simulators, over time. Typically, boundary properties (bottom hole pressure, etc.) are iteratively passed between the simulators, multiple times during each time-step. At the beginning of each Newton-Raphson iteration, the boundary solutions are passed to the complementary simulation, and a new Newton-Raphson iteration is executed with the new boundary conditions. This process is repeated until both simulators converge within a specified tolerance. Additional heuristics are often employed, such as alternative IPR calculations. The third category of coupling, fully implicit coupling, eliminates non-physical oscillations. Fully implicit systems differ from explicit and partially implicit systems in that the fully implicit system is treated as a single simulation. The reservoir, well and surface network dynamic equations are linearized solved simultaneously as one system of equations. However, fully implicit coupled systems have high development costs, and are computationally very expensive. Limitations in explicit and fully implicit coupling has directed efforts towards partially implicit surface-subsurface coupling applications.

While researchers have been successful in finding solutions to partially implicit coupled systems, these solutions typically only apply to specific software packages or propose adjustments that must be made to underlying simulator code. Many existing solutions are computationally expensive and are difficult to implement. In addition, partially implicit simulations are still prone to non-physical oscillation. This study utilized adaptive time-stepping to reduce non-physical oscillations. Our goal was to select optimal time-steps. That is, our adaptive time-stepping scheme would have to both capture periods of transience and reduce total time-steps taken. Therefore, our time-step selection scheme would execute small time-steps during periods of significant transient pressure, and optimize time-steps in steady-state pressure conditions by selecting the largest time-step that would not result in oscillation. In addition, we wanted to utilize a tool that would automatically calculate optimized time-steps. We drew upon control theory to satisfy our desire to employ an automated adaptive time-stepping scheme. Although control theory has been utilized to control time-stepping in reservoir simulation, it has never been used to reduce non-physical oscillations in coupled surface-subsurface. Therefore we sought to explore this possibility. Control theory is the application of engineering and mathematics to study dynamical system behavior. The objective of control theory is to control an output of a system, usually by measuring a complementary input or output and then generating a signal used to change an operating condition. Controllers have been used in many industries for decades, and are found in products such as thermostats, vehicular cruise controls, and modern ovens. Although control theory was first introduced in the 1920’s, control schemes have been adapted in diverse applications including neural networks and gene expression description.

We defined coupling error as the difference between pressures at the explicit coupling point (the tubing head) as calculated by the reservoir simulation (INTERSECT) and the network simulation (PIPESIM) [PIPESIM, INTERSECT]. We implemented a PID controller that measured coupling error and controlled the time-step in a partially implicit coupled system. Prior work had shown that the control of time-stepping successfully reduced error as well as optimized time-step size in numerical methods in mathematics [14, 38], and that control of pressure and saturation change reduced computational costs in reservoir simulation. Our objective was to show that our PID controller was effective in reducing non-physical oscillations in a commercial setting, as well as create a methodology that can easily be implemented in any commercial simulator that allows the user to select the time-step. Our experiments compared rate and pressure behavior associated with a PID controller used to select time steps in Schlumberger's INTERSECT (IX) *Field Management* partially implicit coupled surface-subsurface system, with *Field Management*'s standard time-stepping algorithm. In addition, we performed a sensitivity analysis on the gains of the PID controller as well as tested various error control schemes.

1.1 Literature review

In this section, the development of surface-subsurface coupling and use of PID control in dynamic systems will be reviewed.

There are two separate categories of coupling: advanced well modeling and surface-subsurface coupling. Advanced well modeling involves detailed modeling of completions and other well components, such that the boundary conditions between the well and reservoir are usually specified. However, it is required that the boundary conditions be controlled by constraining certain aspects of well design and operating conditions.

Advanced well models do improve coupling between the well and reservoir, but they do not account for dynamic and complex surface facilities. Surface-subsurface coupling involves the integration of all aspects of production: the reservoir, well and surface facilities. The following review presents a brief history of the broader category of surface-subsurface coupling.

Surface-subsurface coupling has been a topic of interest in simulation for the past four decades. Separate reservoir and network simulators arose as fully implicit simulation was not feasible in early computers and surface-subsurface models. Thus, much of the early surface-subsurface coupling work was focused on explicit and partially implicit coupled simulation. [37] discussed early couplings ideas. [6] formulated a coupling scheme and presented an algorithm applied to three example fields. The simulations included both single phase gas and two phase gas and water reservoirs (tight sands) as well as multiple wellbores and multiple-trunk, separate gathering systems. [6] recognized that well deliverability curves do not remain fixed between time-steps, and so devised a coupling algorithm that forced reservoir, well and surface network boundary conditions to converge within a tolerance on the time step level. This scheme forced convergence of production rate in the well and surface network by first assuming a well flow rate, then by computing the pressure at various nodes in the surface network in order to obtain a tubing head pressure. For the single-phase system, bottom-hole pressures were then calculated from both the Smith equation [21] (wellbore) and the Generalized Flow Equation (surface network) [21]. Subsequently calculated well flow rates were compared. If the rates did not converge within 0.5%, the process was repeated with a new, manually guessed flow rate.

After convergence was achieved in the well and surface network sub iteration, the bottom-hole pressure solution would be passed to the reservoir simulator as a boundary condition for the next time step. This nascent coupling algorithm relaxed constant drainage area assumptions, allowing for improved simulation of infill drilling. [36], and [9] improved upon original coupling experiment. Improvements included use of a large three phase reservoir [11], as well as the implementation of an estimated middle time-step reservoir pressure in wellbore deliverability calculations [29]. The use of middle time step pressures was a means to better represent the well deliverability curve across the time-step. However, the authors used the Hagedorn-Brown [41] correlation to calculate pressure drop through well tubing. This coupling system is therefore only applicable to vertical wellbores. [30] improved upon coupling efficiency by utilizing a preconditioner at the beginning of each time-step. This provided estimated well subdomain boundary conditions at the beginning of the new time-step, and increased the balancing rate between well and reservoir during the first Newton-Raphson iteration of the simulation. This resulted in flow rates that were in better agreement with fully implicit solutions. Applying preconditioners was first used in reservoir simulation when local grid refinement was tested, and [30] illustrated that numerical methods can be extended to improve coupling. [3] applied the use of a preconditioner to a fully implicit system. The resulting simulation was superior to fully implicit coupled simulations with regards to CPU time, however, many aspects of the model were simplified and total CPU time was still prohibitive to commercial application.

Surface-subsurface coupling algorithms became available in commercial simulators in the late 1980s. The commercial simulators were coupled at the wellhead, allowing for real-time operational optimization, and were coupled within the time step at the Newton-Raphson iteration level. Schlumberger's ECLIPSE 200 is an example of a simulator that incorporated these improvements. [23] successfully implemented coupling schemes into a commercial, compositional reservoir. They tested both a fully implicit and partially implicit scheme. In their implicit simulation, the network, tubing head, reservoir and well were solved simultaneously, by treating the well and all downstream components as additional grid blocks in the reservoir. One example simulation utilized 15,300 gridblocks of radial geometry and of various size. This simulation converged at all time steps (illustrating unconditional stability), and satisfactorily matched with historical production rates from the real field analogy. In their partially implicit coupling scheme test, the simulator did not converge when the bottom-hole pressure was unconstrained. When the bottom-hole pressure was constrained, the simulation did converge, however not to the correct solution. A second example simulation executed by [23] illustrated the benefits and drawbacks of using analytical methods for modeling multiphase fluid in well tubing strings, as compared to the use of hydraulic tables. [13] created a set of heuristic controls and a user interface that improved coupling within a complex reservoir and network simulation for various partially implicit schemes. Their interface allowed the use of different software packages, enabled the integration of various fluid models as well as allowed utilization of external optimization algorithms. In addition, the authors created a tightly coupled, *Iteratively Lagged* scheme for single reservoir systems, and a looser coupling for systems of multiple reservoirs. In order to improve convergence and balance coupled systems, their set of heuristics included various rules designed to impose global and local constraints on system components. These constraints were applied to bottom-hole pressures, tubing head pressures, and flow rates in their partially implicit coupled surface-subsurface system. The authors simulated multiple reservoirs tied to both a single production facility and a more complex gathering system with separation and water reinjection. [18, 19, 22] did similar coupling experiments. [5] improved upon earlier preconditioned fully implicit simulation by assuming steady-state surface network behavior. In addition to surface-subsurface modeling, the authors utilized well domain decomposition and advanced well modeling. This method improved the coupling efficiency of the well and network such that it rivaled existing partially implicit methods. The authors focused on computational efficiency of their model, and there is little discussion upon the effects of their coupling scheme on production rates and pressure behavior over time. The motivated reader can find additional examples of early coupled simulation in [19].

[17] formulated time-lagged and partially implicit coupling schemes that utilize a well subdomain in IPR calculations. The subdomains were selected with an algorithm that considered cells containing completions, as well as cells with larger pressure gradients. The object of subdomain cell selection was to identify the cells that would facilitate significant volumetric flow as well as experience dynamic changes in pressure and flux across time. The subdomain was treated as an independent reservoir, where boundary cell fluxes were used as boundary conditions. The use of the subdomain IPR significantly

reduced oscillation in the authors' coupling scheme. However, large coupling time steps led to error. Unnecessarily large subdomains resulted in prohibitive CPU time. [16] built a simulator framework in Schlumberger's INTERSECT and PIPESIM [31, 32, 33, 34, 35] and implemented *Field Management* (FM). *Field Management* was introduced as a simulation workflow to optimize field development, surface-subsurface coupling as well as aid in surface network design and de-bottlenecking. [12] studied coupling schemes in Schlumberger's *Field Management*, and compared results to a fully implicit coupled system. [12] also determined the optimal number of Newton-Raphson iterations in partially implicit coupling, and that Schlumberger's *Field Management* coupled system production matched well with a fully implicit simulation in MRST [24]. Recently, the authors in [39] used a combination of analytical scaling and the fast marching method (FFM) to reduce oscillations in a coupled model. Their work concisely describes the coupling problem. For readers interested in FMM, [40] provides a study relevant to [39]. [4] provides additional review of modern coupling approaches. The next part of this review will focus on general PID control theory and control specific to our work.

PID control has been utilized in the aerospace, chemical, automotive, aviation and consumer electronics industries for decades. Despite this, there has been limited use of PID control in production and reservoir systems. Although PID control has not been used in surface-subsurface coupled systems prior to this body of work, PID control has been implemented in both slug control and reservoir simulation. [27] created a feedback PID control system to effectively reduce slugging while minimizing backpressure exerted on the surface network and well upstream of a riser control valve. By minimizing pressure exerted by the control valve, the authors were able to increase production during periods of slugging. Both PI and PID control was used. A relay auto-tuning approach was used to optimize the gains for the PI and PID controllers. The authors identified the valve position that minimized back pressure by analyzing controlled riser pressure stability across time in an OLGA simulation. Typical control valves used in slug control schemes operate on open loop configurations. Despite slug control's well-defined problems surround backpressure and production loss, most investigations focused solely on reducing slugs in production systems. The use of closed-loop PID control allowed the authors to optimize production in addition to controlling slugs. In reservoir simulation, [1] used a PID controller to perform adaptive time stepping in a two-phase reservoir. The PID controller monitored change in saturation and pressure across time-steps and controlled the size of the time-step. The authors' PID controlled adaptive time-stepping algorithm reduced computation time, total number of time steps, and total number of Newton-Raphson iterations. All three performance parameters varied as a function of simulation time. For an experiment designed to simulate five years, the PID reduced computation time by over 30%, as compared with traditional time-stepping methods. For all simulations, the PID and traditional simulation had nearly identical production profiles. The control of the time-step also highlighted the importance of time-step size selection in reservoir simulation convergence, accuracy and computation time [1]. [20] used a first order predictor-corrector method to improve upon time-step selection methods in reservoir simulation. This was one of the first explorations into adaptive time-stepping in reservoir engineering The author considered a set of moderately stiff ordinary differential equations (ODE's) and associated mean-square norm definition for truncation error. A well-established method to solve ODEs was modified to integrate Navier-Stokes equations in order to accommodate a steam injection simulator. The resulting adaptive time-stepping scheme reduced computational effort and eliminated divergence and oscillation.

The use of PID controlled adaptive time-stepping in numerical simulation is well established and of great importance. [14] used automatic control to perform numerical integration of ODE's. The controller reduced the computational cost of the numerical algorithm while satisfying accuracy requirements. The controller was designed to measure local truncation error and minimize both number of time-steps and associated time-step changes. Experiments were performed using proportional integral (PI) control for seven different ODE's. A smooth time-step response was achieved. In effect, the controller was able to select the maximum time-step that kept changes in accuracy within a user defined tolerance. [7] described the use of PID control in time-stepping problems, and [Valli et al. (2002)] utilized PID time-step control in the simulation of a 2D coupled viscous flow and heat transfer problem.

PID control has been shown to be an effective method to reduce error and computational time in numerical simulators, yet control methods for such purposes are not common in oil and gas applications. Time-step control strategies have not been implemented in coupled simulations: this study intends to bridge that gap. The following body of work yields insight into the effect of time-step selection in partially implicit coupled simulation. This is a neglected area of study, and our work shows promise as a robust and straightforward means to further improve coupled simulation performance.

1.2 Objective of the study

This study assesses the ability of PID control to reduce oscillation in a partially explicit coupled simulation. PID control is used to perform adaptive time-stepping in a widely used commercial simulator, INTERSECT. The objectives of the study were as follows:

1. Determine the ability of a PID controller to reduce non-physical oscillations in a partially implicit coupled system by controlling time-steps.
2. Perform a sensitivity analysis to identify optimal controller gains for our reservoir and surface network model, in order to determine which PID controller best reduces oscillations and manages computation time.
3. Determine the PID controller's ability to compete (with respect to simulation time and cumulative production error) with Schlumberger's default coupling settings.
4. Evaluate different error measurement schemes to understand how to implement PID control in a field with multiple wells.
5. Identify future work to improve PID control in time-stepping for coupled simulation.

1.3 Thesis organization

Chapter two illustrates flow equations and other theoretical background required to describe both surface and subsurface simulation used in this study. Configured settings and model descriptions used in the chosen surface and subsurface simulators, PIPESIM and INTERSECT, are discussed.

Chapter three provides background on *Field Management*, *Field Management* coupling schemes and general control theory. The PID characteristic equation is derived, and the PID time-stepping algorithm used in this study is described in detail. Gain selection is discussed.

Chapter four discusses results. Coupling performance parameters needed to compare *Field Management*'s coupled simulation to PID controlled adaptive time-stepping are discussed. The base case coupled simulation, sans PID control, is presented. Definition of coupling error is introduced. Results of PID controlled adaptive time-stepping are compared to *Field Management*'s base case. Manual PID gain sensitivity analysis is described in detail to understand how coupling performs using various controller gain settings. A field with two producers is simulated, and various error-input schemes are tested to understand the impact of well configurator on simulator performance. Error analysis is performed to elucidate *Field Management*'s handling of coupling operations, and we propose our own set of coupling guidelines. We perform stress tests in *Field Management* in order to highlight simulation issues. In addition, a manually optimized PID controller is used to motivate additional work needed to be done.

Chapter 5 discusses general conclusions and proposes future work.

CHAPTER II

SURFACE AND SUBSURFACE MODELING

The simulations carried out in this study utilized Schlumberger's INTERSECT and PIPESIM software. INTERSECT's *Field Management* application was used to perform partially implicit coupling between INTERSECT and PIPESIM. INTERSECT is a modern, multiple feature reservoir simulator developed jointly by Chevron, Total and Schlumberger. The software is trademarked and licensed by Schlumberger. Schlumberger's PIPESIM software was used as the underlying network simulator in this study. PIPESIM is a steady-state, multiphase flow simulator. PIPESIM models fluid from the wellbore to surface facilities and pipelines. *Field Management* software was designed to integrate diverse oilfield simulation and operating constraints, in order to perform field-wide optimization.

In this chapter, we give a brief introduction to both simulators by stating the underlying equations and computational aspects of the software.

2.1 Black oil reservoir model

The object of the study was to test the concept of PID control, and thus the black oil fluid model was selected in INTERSECT. The black oil model used in INTERSECT features three phase flow in porous media. The black oil model assumes that fluid (gas, oil, water) properties are solely a function of pressure. That is, temperature and fluid composition are not considered in bulk fluid property calculations. Additionally, the black oil model assumes gas is soluble in oil, but has not in water. When reservoir pressure is above bubble point pressure, p_b, the reservoir is undersaturated. There is no free gas at reservoir conditions:

$$S_w + S_o = 1, \qquad p_b < p_o, \qquad R_s = 0$$

Free gas may be present in a black oil reservoir when reservoir pressure drops below the bubble point pressure. For a saturated reservoir:

$$S_w + S_o + S_g = 1, \qquad p_o \leq p_b, \qquad R_s > 0$$

Next, we introduce the PDE equations of the black oil model reservoir and the surface network.

2.2 Black oil model derivation

Three phase flow in porous media, as employed in the underlying simulation, will be discussed in this chapter. [10] was used as a guide. The concept of mass balance applies to each component c in the three phase system:

$$(mass_{in})_c - (mass_{out})_c + S_c = (mass_{accumulation})_c \qquad (2.1)$$

For each component c (oil, water, gas), $mass_{in}$ and $mass_{out}$ are determined by positive and negative fluxes across each face of a gridblock in simulation for a given period of time. Accumulation (the RHS of (2.1)) is a function of changing fluid saturation as well as the compressibility of the individual component. The source/sink component, S_c, is classically defined as mass generation and depletion of component *c*. There are no chemical reactions considered in the black oil model. Wells (injectors and producers) model mass generation/depletion in the black oil model. For components *o*, *w*, and g, Darcy's law states:

$$\underset{U}{\rightarrow}{}_{c} = \beta_c \ \frac{kk_{rc}}{\mu_c}\left(\underset{\nabla}{\rightarrow} p_c - \gamma_c \underset{\nabla}{\rightarrow} Z\right) \tag{2.2}$$

After redefining mass flux using Darcy's velocity, Darcy's law (2.2) is substituted into mass conservation equations to obtain the following. For oil, water and gas:

$$\nabla\left[\frac{\rho_o k k_{ro}}{\mu_o}\ (\nabla p_o - \rho_o g \nabla z)\right] = \frac{\partial[\rho_o \phi S_o]}{\partial t} + q_o \tag{2.3}$$

$$\nabla\left[\frac{\rho_w k k_{rw}}{\mu_w}\ (\nabla p_w - \rho_w g \nabla z)\right] = \frac{\partial[\rho_w \phi S_w]}{\partial t} + q_w \tag{2.4}$$

$$\nabla\left[\frac{\rho_{go} k k_{ro}}{\mu_{go}}\ (\nabla p_o - \rho_o g \nabla z) + \frac{\rho_g k k_{ro}}{\mu_g}\ \left(\nabla p_g - \rho_g g \nabla z\right)\right] = \frac{\partial\left[(\rho_{go} \phi S_o + \rho_g \phi S_g\right]}{\partial t} + q_g \tag{2.5}$$

The RHS's of equations (2.3-2.5) comprise of the well source/sink term, $q_{o,w,g}$, and the accumulation term. Noting gas dissolution in oil, ρ_{go} is the partial density of gas in the oil phase. Noting that R_s is the dissolved gas-oil ratio, R_s is equivalent to the volume of gas $V_{g,s}$ (at STP) dissolved in a unit volume of oil, and that $V_{o,s}$, at stock tank conditions.

$$R_s = \frac{V_{g,s}}{V_{o,s}} \qquad (2.6)$$

Additional relationships are required to describe capillary pressures and fluid saturation constraints for the general case:

$$P_{cow} = P_o - P_w \qquad (2.7)$$

$$P_{cgo} = P_g - P_o \qquad (2.8)$$

$$P_{cgw} = P_{cgo} - P_{cow} \qquad (2.9)$$

$$S_w + S_g + S_o = 1 \qquad (2.10)$$

Where equations 2.8-2.10 illustrate the oil-water, gas-oil and gas-water capillary pressures, respectively. Additionally, oil and gas volumes can be substituted with weight and density:

$$V_o = \frac{W_o}{\rho_o} \qquad (2.11)$$

$$V_g = \frac{W_g}{\rho_g} \qquad (2.12)$$

Substituting equations 2.12 and 2.13 into 2.6:

$$R_s = \frac{V_{g,s}}{V_{o,s}} \qquad (2.13)$$

$$R_s = \frac{W_g \rho_o}{W_o\, \rho_g} \qquad (2.14)$$

The formation volume factor, *B*, is defined as the ratio of the volume of one phase at reservoir conditions to the volume of the same phase at standard conditions.

$$B = \frac{V_{phase\,c,\ \ reservoir}}{V_{phase\,c,\ \ standard\ conditions}} \tag{2.15}$$

Substituting for weight and density:

$$\rho_g = \frac{\rho_{g,standard\ conditions}}{B_g}\ ,\ \rho_w = \frac{\rho_{w,standard\ conditions}}{B_w} \tag{2.16}$$

Due to dissolved gas in the oil, there are two separate definitions for density: that of the oil phase and that of the gas dissolved in the oil phase.

$$\rho_o = \frac{R_s\rho_{g,standard\ conditions}+\rho_{o,standard\ conditions}}{B_o} \tag{2.17}$$

$$\rho_{gas\ in\ oil} = \frac{R_s\rho_{g,standard\ conditions}}{B_o} \tag{2.18}$$

Substituting equations 2.16 and 2.18 into 2.3-2.5:

$$\nabla\left[\frac{\rho_o k k_{ro}}{\mu_o B_o}\ (\nabla p_o - \rho_o g \nabla z)\right] = \frac{\partial}{\partial t}\left[\frac{[\rho_o \phi S_o]}{B_o}\right] + q_o \tag{2.19}$$

$$\nabla\left[\frac{\rho_w k k_{rw}}{\mu_w B_w}\ (\nabla p_w - \rho_w g \nabla z)\right] = \frac{\partial}{\partial t}\left[\frac{[\rho_w \phi S_w]}{B_w}\right] + q_w \tag{2.20}$$

$$\nabla\left[\frac{\rho_{go} k k_{ro}}{\mu_o B_o}\ (\nabla p_o - \rho_o g \nabla z) + \frac{\rho_g k k_{ro}}{\mu_{gB_g}}\ \left(\nabla p_g - \rho_g g \nabla z\right)\right] = \frac{\partial}{\partial t}\left[\frac{[R_s\rho_{g,standard}\phi S_o]}{B_o} + \frac{[\rho_{g,standard}\phi S_g]}{B_g}\right] + q_g + q_o R_s \tag{2.21}$$

By identifying the ratio of effective permeability and viscosity as the mobility ratio, λ, and by dividing by density, equations 2.3-2.5 can be re-written as

$$\nabla[\lambda_0\ (\nabla p_o - \rho_o g \nabla z)] = \frac{\partial}{\partial t}\frac{[\phi S_o]}{B_o} + q_o^* \qquad (2.22)$$

$$\nabla[\ \lambda_w(\nabla p_w - \rho_w g \nabla z)] = \frac{\partial}{\partial t}\frac{[\phi S_w]}{B_w} + q_w^* \qquad (2.23)$$

$$\nabla\left[R_s\lambda_0(\nabla p_o - \rho_o g \nabla z) + \ \lambda_g\left(\nabla p_g - \rho_g g \nabla z\right)\right] = \frac{\partial}{\partial t}\left[\frac{[R_s\phi S_o]}{B_o} + \frac{\left[\phi S_g\right]}{B_g}\right] + q_o^* R_s + q_g^* \qquad (2.24)$$

Including equations 2.8-2.11, and applying the nabla operator, mass conservation can be expanded to three dimensions:

$$\frac{\partial}{\partial x}\left[\lambda_0\ \left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial Z}{dx}\right)\right] + \frac{\partial}{\partial y}\left[\lambda_0\left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial Z}{dy}\right)\right] + \frac{\partial}{\partial z}\left[\lambda_0\ \left(\frac{\partial p_o}{\partial z} - \gamma_o \frac{\partial Z}{dz}\right)\right] =$$

$$\frac{\partial}{\partial t}\left(\frac{\phi(1 - S_w - S_g}{B_o}\right) + q_o^* \qquad (2.25)$$

$$\frac{\partial}{\partial x}\left[\lambda_w \left(\frac{\partial p_o}{\partial x} - \frac{\partial p_{cow}}{\partial x} - \gamma_w \frac{\partial Z}{dx}\right)\right] + \frac{\partial}{\partial y}\left[\lambda_w \left(\frac{\partial p_o}{\partial y} - \frac{\partial p_{cow}}{\partial y} - \gamma_w \frac{\partial Z}{dy}\right)\right] + \frac{\partial}{\partial z}\left[\lambda_w \left(\frac{\partial p_w}{\partial z} - \frac{\partial p_{cow}}{\partial z} - \gamma_w \frac{\partial Z}{dz}\right)\right] = \frac{\partial}{\partial t}\left(\frac{\phi S_w}{B_w}\right) + q_w^* \qquad (2.26)$$

$$\frac{\partial}{\partial x}\left[\lambda_0 R_s \left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial Z}{dx}\right)\right] + \frac{\partial}{\partial y}\left[\lambda_0 R_s \left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial Z}{dy}\right)\right] + \frac{\partial}{\partial z}\left[\lambda_0 R_s \left(\frac{\partial p_o}{\partial z} - \gamma_o \frac{\partial Z}{dz}\right)\right] + \frac{\partial}{\partial x}\left[\lambda_g \left(\frac{\partial p_g}{\partial x} - \frac{\partial p_{cgo}}{\partial x} - \gamma_g \frac{\partial Z}{dx}\right)\right] + \frac{\partial}{\partial y}\left[\lambda_g \left(\frac{\partial p_g}{\partial y} - \frac{\partial p_{cgo}}{\partial y} - \gamma_g \frac{\partial Z}{dy}\right)\right] + \frac{\partial}{\partial z}\left[\lambda_g \left(\frac{\partial p_g}{\partial z} - \frac{\partial p_{cgo}}{\partial z} - \gamma_g \frac{\partial Z}{dz}\right)\right] = \frac{\partial}{\partial t}\left(\frac{R_s \phi (1 - S_w - S_g}{B_o} + \frac{\phi S_g}{B_g}\right) + R_s q_o^* + q_g^* \qquad (2.27)$$

Where $\gamma_{o,w,g}$ is equivalent to the product of the gravitational constant, g, and $p_{o,w,g}$ respectively.

2.3 Discretization of flow equations

The following discretization of the previously illustrated analytical flow equations will be executed using a block-centered Cartesian grid. As a note, these equations describe the basic premises on which Schlumberger's INTERSECT is built. INTERSECT has additional, advanced physics. However, we utilized the black oil model, and thus this description is salient to our experiment. Some items, such as numerical solvers, are not discussed. The reader should refer to the INTERSECT manual for further information.

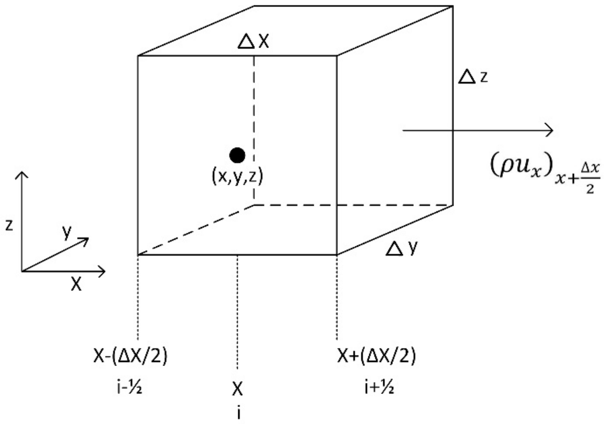

Figure 2.1: Flow across block-centered grid block in +x direction

In the reservoir simulation, flow can only move perpendicular to the six planes as illustrated in Figure 2.1. Mass flux in the x, y and z direction can be described as:

$$(\rho u_x)_{x\pm\frac{\Delta x}{2}} \qquad (2.28)$$

$$(\rho u_y)_{y\pm\frac{\Delta y}{2}} \qquad (2.29)$$

$$(\rho u_z)_{z\pm\frac{\Delta z}{2}} \qquad (2.30)$$

Substituting with component mass flux, $\dot{m}_c$, we can recast quantities from the simplified mass balance:

$$(mass_{in})_c = \left[(\dot{m}_{cx}A_x)_{x-\frac{\Delta x}{2}} + (\dot{m}_{cy}A_y)_{y-\frac{\Delta y}{2}} + (\dot{m}_{cz}A_z)_{z-\frac{\Delta z}{2}}\right]\Delta t \qquad (2.31)$$

$$(mass_{out})_c = \left[(\dot{m}_{cx}A_x)_{x+\frac{\Delta x}{2}} + (\dot{m}_{cy}A_y)_{y+\frac{\Delta y}{2}} + (\dot{m}_{cz}A_z)_{z+\frac{\Delta z}{2}}\right]\Delta t \qquad (2.32)$$

In the mass conservation equations above, $A_{x,y,z}$ is the area perpendicular to flux with respect to the Cartesian dimension in the subscript. Transmissibility and pressure gradients are likewise discretized. The x-y plane is assumed to be parallel with the horizontal. For water conservation in the x,y and z direction:

$$\frac{\partial}{\partial x}\left[\lambda_w\ \left(\frac{\partial p_o}{\partial x}-\frac{\partial p_{cow}}{\partial x}-\gamma_w\frac{\partial Z}{dx}\right)\right]=$$

$$\frac{1}{\Delta x_i}\left[\lambda_{wi+\frac{1}{2},j,k}\frac{p_{oi+1,j,k}-p_{oi,j,k}}{\Delta x_i^+}+\lambda_{wi-\frac{1}{2},j,k}\frac{p_{oi-1,j,k}-p_{oi,j,k}}{\Delta x_i^-}\right] \quad (2.33)$$

$$\frac{\partial}{\partial y}\left[\lambda_w\ \left(\frac{\partial p_o}{\partial y}-\frac{\partial p_{cow}}{\partial y}-\gamma_w\frac{\partial Z}{dy}\right)\right]=$$

$$\frac{1}{\Delta y_i}\left[\lambda_{wi,j+\frac{1}{2},k}\frac{p_{oi,j+1,k}-p_{oi,j,k}}{\Delta y_i^+}+\lambda_{wi,j-\frac{1}{2},k}\frac{p_{oi,j-1,k}-p_{oi,j,k}}{\Delta y_i^-}\right] \quad (2.34)$$

$$\frac{\partial}{\partial z}\left[\lambda_w\ \left(\frac{\partial p_w}{\partial z}-\frac{\partial p_{cow}}{\partial z}-\gamma_w\frac{\partial Z}{dz}\right)\right]=\frac{1}{\Delta z_i}\left[\lambda_{wi,j,k+\frac{1}{2}}\frac{p_{oi,j,k+1}-p_{oi,j,k}}{\Delta z_i^+}+\lambda_{wi,j,k-\frac{1}{2}}\frac{p_{oi,j,k-1}-p_{oi,j,k}}{\Delta z_i^-}-\right.$$

$$\left.\lambda_{wi,j,k+\frac{1}{2}\gamma wi,j,k+\frac{1}{2}}\frac{z_{i,j,k+1}-z_{i,j,k}}{\Delta z_i^+}-\lambda_{wi,j,k-\frac{1}{2}}\gamma_{wi,j,k-\frac{1}{2}}\frac{z_{i,j,k-1}-z_{i,j,k}}{\Delta z_i^-}\right] \quad (2.35)$$

Equations 2.33-2.35 are collected to obtain the discretization of Darcy's water conservation:

$$\frac{\partial}{\partial x}\left[\lambda_w\ \left(\frac{\partial p_o}{\partial x}-\frac{\partial p_{cow}}{\partial x}-\gamma_w\frac{\partial Z}{dx}\right)\right]+\frac{\partial}{\partial y}\left[\lambda_w\ \left(\frac{\partial p_o}{\partial y}-\frac{\partial p_{cow}}{\partial y}-\gamma_w\frac{\partial Z}{dy}\right)\right]+\frac{\partial}{\partial z}\left[\lambda_w\ \left(\frac{\partial p_w}{\partial z}-\right.\right.$$

$$\left.\left.\frac{\partial p_{cow}}{\partial z}-\gamma_w\frac{\partial Z}{dz}\right)\right]=\frac{1}{\Delta x_i}\left[\lambda_{wi+\frac{1}{2},j,k}\frac{p_{oi+1,j,k}-p_{oi,j,k}}{\Delta x_i^+}+\lambda_{wi-\frac{1}{2},j,k}\frac{p_{oi-1,j,k}-p_{oi,j,k}}{\Delta x_i^-}\right]+$$

$$\frac{1}{\Delta y_i}\left[\lambda_{wi,j+\frac{1}{2},k}\frac{p_{oi,j+1,k}-p_{oi,j,k}}{\Delta y_i^+}+\lambda_{wi,j-\frac{1}{2},k}\frac{p_{oi,j-1,k}-p_{oi,j,k}}{\Delta y_i^-}\right]+\frac{1}{\Delta z_i}\left[\lambda_{wi,j,k+\frac{1}{2}}\frac{p_{oi,j,k+1}-p_{oi,j,k}}{\Delta z_i^+}+\right.$$

$$\left.\lambda_{wi,j,k-\frac{1}{2}}\frac{p_{oi,j,k-1}-p_{oi,j,k}}{\Delta z_i^-}-\lambda_{wi,j,k+\frac{1}{2}\gamma wi,j,k+\frac{1}{2}}\frac{z_{i,j,k+1}-z_{i,j,k}}{\Delta z_i^+}-\lambda_{wi,j,k-\frac{1}{2}}\gamma_{wi,j,k-\frac{1}{2}}\frac{z_{i,j,k-1}-z_{i,j,k}}{\Delta z_i^-}\right]$$

$$(2.36)$$

where (i,j,k) are coordinate indices analogous to (x,y,z) in Cartesian coordinates. The subscript $i \pm \frac{1}{2}$ describes the locations of the block faces whose planes are perpendicular to flow along the x axis. Transmissibility and pressure gradients are calculated at each gridblock interface to account for flow in and out of the cell.

And by similar analogy, the time discretized accumulation term for the water phase:

$$\frac{\partial}{\partial t}\frac{\phi S_w}{B_w} = S_w^n(b_w^{n+1}\phi' + \phi' b_w')\Delta p_w + (\phi^{n+1} b_w^{n+1})\Delta S_w \qquad (2.37)$$

where $\Delta S_w = \frac{S_w^{n+1} - S_w^n}{\Delta t},\ \Delta p_w = \frac{p_w^{n+1} - p_w^n}{\Delta t},\ \phi' = \frac{\phi^{n+1} - \phi^n}{p_w^{n+1} - p_w^n},\ b_w' = \frac{b_w^{n+1} - b_w^n}{p_w^{n+1} - p_w^n}$

and $b_w = \frac{1}{B}$.

The superscript n indicates the value of the properties at the time-step n. The superscript n+1 illustrates the value of the same properties at the next time-step.

INTERSECT utilized Peaceman's equation [28] in our structured grid simulation. The wellbores in our simulation were vertical and penetrated the full length of the completed gridblock through its center. None of the wells were fractured. The sink/source term for water is defined using Peaceman's equation:

$$q_w^* = WI_w({p_w}_{gbc}^{n+1} - p_{wf}) \quad (2.38)$$

Where WI_w is the well index describing the water phase, and ${p_w}_{gbc}$ is the water phase pressure in the gridblock the well is completed in. The well index is defined as:

$$WI_w = -\frac{2\pi k_{rw}\sqrt{k_x k_y h}}{\mu_w B_w\left[\ln\left(\frac{r_o}{r_w}\right)+s\right]} \qquad (2.39)$$

Where s is the skin factor, μ_w is water viscosity, B_w is the water formation volume factor, and k_x, k_y are permeability in the x and y direction, respectively. r_w is the wellbore radius, and the equivalent radius, r_o, is defined as:

$$r_o = \frac{\sqrt{\left[\left(\frac{k_y}{k_x}\right)^{1/2}(\Delta x)^2\right]+\left[\left(\frac{k_x}{k_y}\right)^{1/2}(\Delta y)^2\right]}}{\left[\left(\frac{k_y}{k_x}\right)^{1/4}+\left(\frac{k_x}{k_y}\right)^{1/4}\right]} \qquad (2.40)$$

This set of reservoir dynamic equations are adequate to describe and introduce reservoir simulation for the purpose of this thesis.

2.4 INTERSECT & Schlumberger time-step selection settings

Schlumberger generated and shared a coupling file deck. The deck included both INTERSECT and PIPESIM files. This section will discuss settings implemented in the INTERSECT deck. Files originated by Schlumberger included **reservoir_edits.ixf* and **IX.ixf*. In the later file, the fully implicit numerical solution method was chosen. This was specified using the *TimeDiscretizationMethod* keyword within the *TimeStepSolution* child node. Time step selection controls were defined in the *TimestepSizingControls* child node. For initial experiments, the following properties were relaxed in the aforementioned node. *TargetCellPressureChange* was specified at 6,000 PSIA. *TargetTimeTruncationError,* *TargetGasSaturationChange*,*TargetOilSaturationChange*, and *TargetWaterSaturationChange* were all set to 0.8. This was to eliminate INTERSECT *hrep* overrides during the time-step selection process. The *hrep* is a forced half-time-step, utilized to help ensure linear and nonlinear convergence controls were satisfied. The half-time-step was forced when the previously mentioned *TimestepSizingControls* property IDs were violated. To balance the relaxed *TimestepSizingControls* settings, we improved convergence by setting *MaxNewtons* to 400. *MaxNewtons* is the property ID that specifies the maximum number of Newton-Raphson iterations INTERSECT can execute in order to achieve convergence for a given time-step. This value was large and arbitrary: the number of reservoir Newton-Raphson iterations per time-step rarely exceeded 40 iterations. All other *TimestepSizingControls* property IDs expressed default values. This ensured we conducted "apples to apples" comparison. However, we conducted later experiments to compare simulation performance of Schlumberger's default coupling settings (no relaxed settings) to PID control with relaxed *TimeStepSizingControls* settings. Important results with respect to PID performance are discussed. Field units were selected. All other **IX.ixf* nodes were specified as Schlumberger's original coupling file deck. The **reservoir_edits.ixf* file solely specified the method used to calculate the tubing head pressure in INTERSECT. The sub node *TubingHeadPressureCalculationOptions* was defined using *LiftCurveStabilization*.

2.5 Network multiphase flow used in coupling

PIPESIM is a multiphase flow simulator capable of simulating the wellbore as well as surface network and pipelines. As mentioned in the introduction, PIPESIM was used as the surface simulation software package. In standalone PIPESIM simulation, reservoir pressure is considered constant. Schlumberger's *Field Management* facilitates coupling when PIPESIM is used in conjunction with INTERSECT to perform rock to pipeline simulation. The black oil reservoir model in INTERSECT results in multiphase flow in the coupled wellbore and surface facility. PIPESIM employs various correlations to ensure multiphase flow is properly modeled. All following theory and formulation of multiphase flow can be found in [63].

PIPESIM operates by calculating pressure drops from the wellbore to surface network, and its objective is to determine deliverability of wells as constrained by well configuration, separator conditions, and pipeline pressure. Pressure loss through a production system can be modeled as follows:

$$p_{wf} = p_{sep} + \Delta p_v + \Delta p_t + \Delta p_c \qquad (2.41)$$

where p_{wf} is the flowing bottomhole pressure, and p_{sep} is the operating pressure of the separator. Δp_v, Δp_t, and Δp_c are pressure loss through valves, pressure loss through tubing and pressure loss though the choke, respectively. Various control schemes are used to regulate pressure at the separator. Schemes include use of control valves and pressure relief valves. A choke at surface was employed to manually control the pressure at the well head as well as fluid flows in separation equipment.

Network simulation discretizes the wellbore/pipelines into segmented control volumes and operates upon the same principal of mass balance as reservoir simulation:

$$(mass_{in})_c - (mass_{out})_c + S_c = (mass_{accumulation})_c \qquad (2.42)$$

Accumulation within pipe control volumes is impacted by holdup. Considering holdup, accumulation within pipeline control volumes can be described using the following equations:

$$\widetilde{ql_{\imath n}} - \widetilde{ql_{out}} = control\ volume * \frac{\partial}{\partial t}(y_l \rho_l) \quad (2.43)$$

$$q_{\widetilde{gas,\imath n}} - q_{\widetilde{gas,out}} = control\ volume * \frac{\partial}{\partial t}(y_g \rho_g) \quad (2.44)$$

The discretized control volume is analogous to the discretized reservoir gridblock. Holdup is indicated with the variables y_l and y_g. In multiphase flow, the denser phases travel through the pipelines/wellbore at slower speeds than the heavier phases. This results in an uneven distribution of phases and bulk density across the pipeline. This phenomena is known as holdup. For mass conversation in tubing or pipe, the tubing is treated as a set of continuous, cylindrical segments. All phases experience one dimensional flow in the +x direction. Figure 2.4 below illustrates the discretization of tubing/piping.

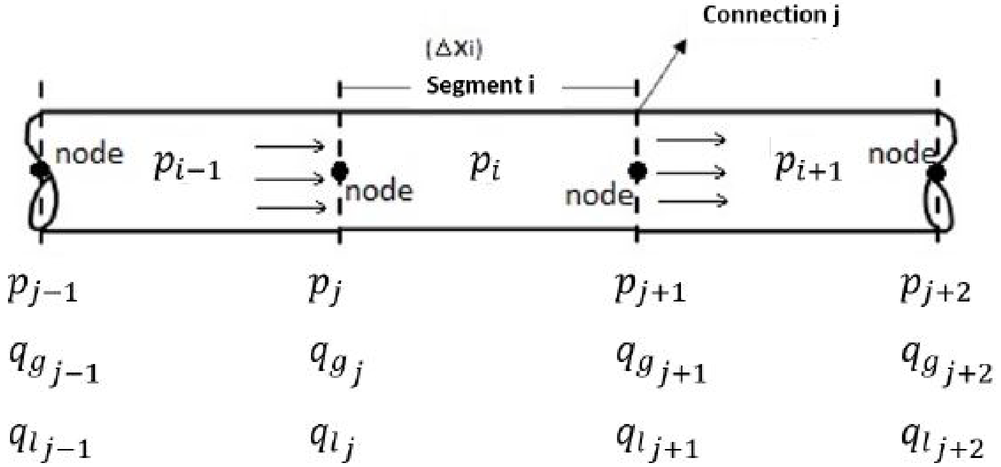

Figure 2.2: Discretized pipeline segment [12]

In figure 2.4, p_i is pressure within segment *i*, ΔX is the length of each segment. p_j, $q_{g\,j}$, and $q_{l\,j}$ are the pressure at interface *j*, the gas flow rate at interface *j*, and the liquid (oil and water) flow rate at interface *j*, respectively. Applying spatial discretization to mass accumulation as illustrated in equations 2.54-2.55 mass conservation for oil, water and gas is illustrated as:

$$\widetilde{q_{wJ}} - q_{\widetilde{wJ+1}} = \Delta x_i A_i * \frac{\partial}{\partial t}(y_w \rho w_s b_w)_i \qquad (2.45)$$

$$\widetilde{q_{oJ}} - q_{\widetilde{oJ+1}} = \Delta x_i A_i * \frac{\partial}{\partial t}(y_o \rho o_s b_o)_i \; (2.46)$$

$$\widetilde{q_{gJ}} - q_{\widetilde{gJ+1}} = \Delta x_i A_i * \frac{\partial}{\partial t}(y_g \rho g_s b_g + R_s y_o \rho o_s b_o)_i \; (2.47)$$

where $\widetilde{q_{o,g,w}}_{J}$ is the mass flow rate at STP, and A_i is the cross sectional area of the control volume in the y-z plane of segment *i*. Then, applying time discretization to equations 2.56-2.58 and dividing by density:

$$\widetilde{q_{wJ}}^{*} - \widetilde{q_{wJ+1}}^{*} = \frac{\Delta x_i A_i}{\Delta t}\left[(y_w b_w)_i{}^{n+1} - (y_w b_w)_i{}^{n}\right] \quad (2.48)$$

$$\widetilde{q_{oJ}}^{*} - \widetilde{q_{oJ+1}}^{*} = \frac{\Delta x_i A_i}{\Delta t}\left[(y_o b_o)_i{}^{n+1} - (y_o b_o)_i{}^{n}\right] \quad (2.49)$$

$$\widetilde{q_{gJ}}^{*} - \widetilde{q_{gJ+1}}^{*} = \frac{\Delta x_i A_i}{\Delta t}\left[(R_s y_o b_o)_i{}^{n+1} - (R_s y_o b_o)_i{}^{n} + (y_g b_g)_i{}^{n+1} - (y_g b_g)_i{}^{n}\right] \quad (2.50)$$

However, PIPESIM modeled steady-state flow in our simulations. Therefore:

$$\widetilde{q_{wJ}}^{*} - \widetilde{q_{wJ+1}}^{*} = \frac{\Delta x_i A_i}{\Delta t}\left[(y_w b_w)_i{}^{n+1} - (y_w b_w)_i{}^{n}\right] \approx 0 \quad (2.51)$$

$$\widetilde{q_{oJ}}^{*} - \widetilde{q_{oJ+1}}^{*} = \frac{\Delta x_i A_i}{\Delta t}\left[(y_o b_o)_i{}^{n+1} - (y_o b_o)_i{}^{n}\right] \approx 0 \quad (2.52)$$

$$\widetilde{q_{gJ}}^{*} - \widetilde{q_{gJ+1}}^{*} = \frac{\Delta x_i A_i}{\Delta t}\left[(R_s y_o b_o)_i{}^{n+1} - (R_s y_o b_o)_i{}^{n} + (y_g b_g)_i{}^{n+1} - (y_g b_g)_i{}^{n} \approx 0\right]$$

$$(2.53)$$

Momentum conservation must also be considered when modeling multi-phase flow in vertical and horizontal pipes. A momentum balance for each discretized segment is as follows:

$$p_i - p_{i+1} = \Delta p_{potential_i} + \Delta p_{friction_i} + \Delta p_{KE_i} \quad (2.54)$$

where $\Delta p_{potential_i}$ is the pressure loss due to potential energy (gravitational forces) in segment i, $\Delta p_{friction_i}$ is the pressure loss due to frictional forces in segment i, and Δp_{KE_i} is pressure loss due to changes in kinetic energy. Pressure loss due to changes in kinetic energy is less significant than pressure loss due to frictional forces and potential energy. After simplifying 2.65 and writing in analytical form:

$$\frac{dp}{dz} = \left[\frac{dp}{dz}\right]_{potential} + \left[\frac{dp}{dz}\right]_{friction} \qquad (2.55)$$

The potential energy term is a function of both gravity and density:

$$\left[\frac{dp}{dz}\right]_{potential} = \frac{g}{g_c}\bar{\rho} sin\theta \qquad (2.56)$$

where θ is the angle of pipe inclination with respect to the origin of the radial axis of the pipe. For a vertically aligned pipe/tubing, $\theta = \frac{\pi}{2}$, and $\theta = 0$ for horizontally aligned pipe/tubing. $\bar{\rho}$ is the average density inside the pipe, and can be described with holdup such that:

$$\bar{\rho} = y_l \rho_l + y_g \rho_g \qquad (2.57)$$

where holdup can be further defined:

$$y_l = \frac{velocity_{liquid}}{velocity_{control\,volume}} \qquad (2.58)$$

holdup is constrained such that:

$$y_l + y_g = 1 \qquad (2.59)$$

Friction loss is further described by expanding the term analytically:

$$\left[\frac{dp}{dz}\right]_{friction} = \frac{2f\rho_m u_m^2}{g_c D} \qquad (2.60)$$

where *D* is pipe diameter, g_c is the gravitational constant, *u_m* is the mixture/bulk velocity, and *f* is the friction factor. ρ_m is bulk density of the flow, and is calculated as such:

$$\rho_m = \lambda_l \rho_l + \lambda_g \rho_g \qquad (2.61)$$

In equation 2.69, λ is the flow fraction, and is defined as:

$$\lambda_l = \frac{q_l}{q_l + q_g} \qquad (2.62)$$

$$\lambda_g = \frac{q_g}{q_l + q_g} \qquad (2.63)$$

$$\lambda_l + \lambda_g = 1 \qquad (2.64)$$

In addition, the mixture velocity is defined as:

$$u_m = u_{sl} + u_{sg} \qquad (2.65)$$

u_{sl} and u_{sg} are superficial velocities and are the ratios of flow rate (gas, liquid) to cross section area, respectively:

$$u_{sl} = \frac{q_l}{A} \qquad (2.66)$$

$$u_{sg} = \frac{q_g}{A} \qquad (2.67)$$

The friction factor is expressed as such below:

$$f = f_n \exp(S) \quad (2.68)$$

where

$$S = \frac{\ln(x)}{-0.0523+3.182\ln(x)-0.8725\,[\ln x]^2+0.01853[\ln x]^4} \quad (2.69)$$

or, equivalently

$$S = \ln(2.2x - 1.2)\ for\ 1 < x < 1.2 \quad (2.70)$$

and

$$x = \frac{\lambda_l}{y_l^2} \quad (2.71)$$

The nonslip friction factor, f_n, is a function of both pipe roughness and the Reynolds number. The Hagedorn-Brown correlation is superior in that it can determine holdup for various flow regimes. In PIPESIM, the modified Hagedorn-Brown [41] correlation was used to determine holdup in vertical pipe. The Hagedorn-Brown correlation is limited to flow in vertical piping. PIPESIM utilized the Beggs and Brill correlation [2] to calculate holdup and identify flow regimes for inclined piping. [62] can be referenced for further background.

2.6 PIPESIM settings

This section will discuss settings implemented in the PIPESIM deck, as provided by Schlumberger. No settings were changed for this experiment. The fluid model was specified as black oil, the separator as single stage, the viscosity model as Newtonian, and an offshore temperature model. There are many additional settings specified, however they are all standard configurations and it is not practical to further define them. As a note, the *OpenPipesim.exe* application was used to initiate the simulation and begin coupling. To access *OpenPipesim*, the file location of *OpenPipesim.exe* was cued in the command window. INTERSECT and PIPESIM must be running prior to starting *OpenPipesim*. After *OpenPipesim* has been cued, INTERSECT is then run with the desired file deck. *OpenPipesim* creates a channel between INTERSECT and PIPESIM, and facilitates the balancing actions as well as other types of communication. *OpenPipesim* also initiates the PIPESIM simulation. To ensure the proper network files are accessed by PIPESIM, the locations of the said files must be specified in the high level **.afi* file. In addition, the coupling server and port must be specified in the **.afi* file. By default, the port is 9010.

2.7 Coupling introduction

As discussed during the introduction, there are three general types of surface-subsurface coupling: explicit, partially implicit, and fully implicit. These three types of coupling will be reviewed. Figure 2.5 below illustrates a basic, explicit coupling scheme. It can be seen that there is no mechanism to ensure that the network solutions and reservoir solutions converge within a tolerance. Forcing a boundary condition on the reservoir model does not guarantee that the reservoir model's rate and pressure solution will converge to the same values that the network simulator calculated. Reservoir and network solutions tend to diverge. Many complications, heuristics, and tools have been created to improve the following three general methodology. This experiment's aim was to provide a solution (automatic time-stepping) that could be easily implemented in any system. The down-side to many existing solutions is that they involve complicated IPR calculations as well as modified simulation. These improvements, though effective, are typically difficult or impossible to implement in commercial software.

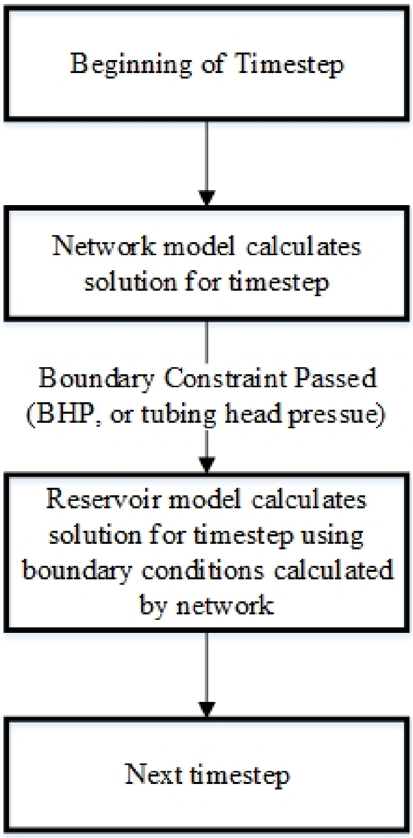
Beginning of Timestep
Network model calculates solution for timestep
Boundary Constraint Passed (BHP, or tubing head pressue)
Reservoir model calculates solution for timestep using boundary conditions calculated by network
Next timestep

Figure 2.3: Explicitly coupled surface-subsurface system coupling algorithm

INTERSECT’s *Periodic* balancing action is analogous to explicit coupling. The *periodic* mode results in lower computational effort requirements. However, this computational benefit is offset with lower simulation accuracy. As a result of diverging surface and subsurface solutions, the coupled simulation yields rates that are significantly different than that of the real, physical system. The *periodic* mode also couples at predetermined time-steps. Therefore, it is impossible to implement an adaptive time stepping algorithm while using the *periodic* mode. The *periodic* mode of coupling is not recommended under any circumstance where the reservoir experiences changing pressure regimes. While still utilizing two separate simulators, the partially implicit method improves upon the explicit method by including algorithms to help the simulators' solutions agree within each time-step. Figure 2.6 below illustrates the general partially implicit coupled scheme.

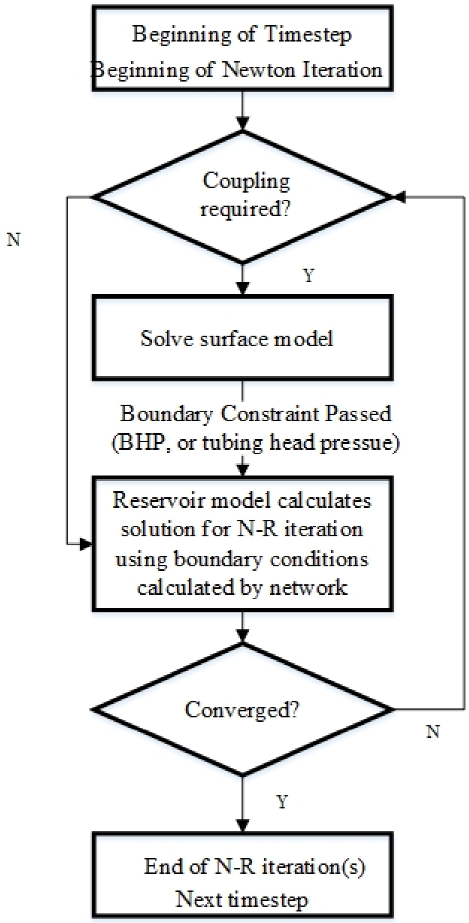
Beginning of Timestep
Beginning of Newton Iteration
Coupling required?
N
Y
Solve surface model
Boundary Constraint Passed
(BHP, or tubing head pressue)
Reservoir model calculates solution for N-R iteration using boundary conditions calculated by network
Converged?
N
Y
End of N-R iteration(s)
Next timestep

Figure 2.4: Partially implicit coupled surface-subsurface system coupling algorithm

The partially implicit coupling method ensures that the surface and subsurface systems are balanced during the Newton-Raphson iterations. These additional computations impose larger computational costs than required by explicit coupling. However, [12] showed that the partially implicit coupling scheme performs just as well as the fully implicit method, without fully implicit computational and developmental requirements. The partially implicit coupling methodology is recommended by INTERSECT's *Field Management*. In this experiment, *Iteratively Lagged* was the selected coupling method. The *Iteratively Lagged* setting is Schlumberger's partially implicit analogy. More will be discussed on Schlumberger's coupling methods in section 2.8. Figure 2.7 below illustrates the fully implicit coupling methodology.

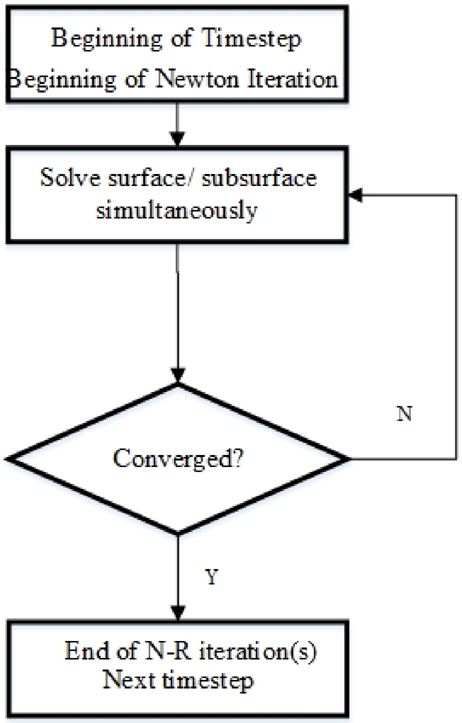
Beginning of Timestep
Beginning of Newton Iteration
Solve surface/ subsurface simultaneously
Converged?
N
Y
End of N-R iteration(s)
Next timestep

Figure 2.5: Fully implicit coupled surface-subsurface system coupling algorithm

The fully implicit coupling methodology treats the surface and subsurface as one system. The systems of equations are solved simultaneously, and convergence of the shared boundary conditions of the reservoir and surface network are guaranteed. The fully implicit method is the most computationally expensive, and the added costs are typically incurred when the Jacobian matrix is solved at each Newton-Raphson iteration. The Jacobian matrix required for a fully implicit coupled system is much larger than the Jacobian matrices found in independent surface and subsurface simulators, respectively. The coupling methodologies described are the three general coupling schemes. As noted, Schlumberger has created coupling schemes that are parallel to the explicit and partially implicit coupling schemes. These coupling schemes are controlled by Schlumberger's *Field Management* software. The following section discusses coupling in *Field Management*.

2.8 Coupling in Schlumberger's *Field Management*

As discussed in section 2.7, Schlumberger does have an explicit coupling scheme. It was noted that the scheme introduced error into the simulation solution. In addition, the explicit scheme produces significant oscillation. Therefore, *Field Management*'s *Periodic* setting was not utilized. Schlumberger's *Iteratively Lagged* setting was chosen. The *Iteratively Lagged* setting is permutation of Schlumberger's *NetBalAct*. *NetBalAct* is Schlumberger's acronym for *Network Balance Action*. *NetBalAct* is the balancing algorithm that Schlumberger has developed specifically for surface and subsurface coupling. *NetBalAct* works within *Field Management* to enable the coupling of the reservoir, well and surface pipelines and facilities. Figure 2.8 illustrates *NetBalAct*'s general balancing algorithm.

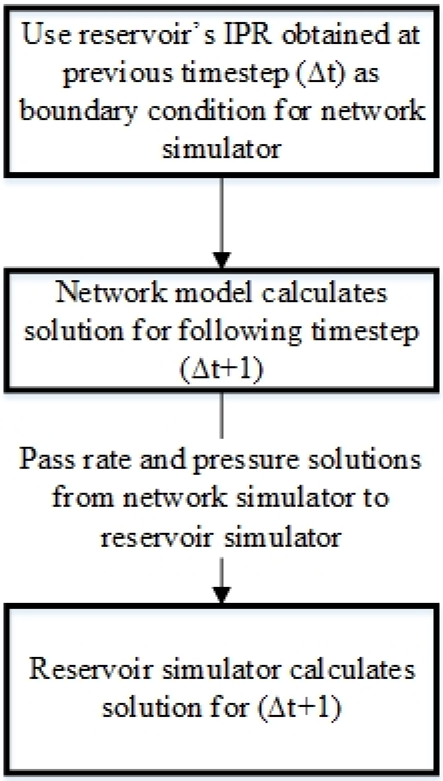
Use reservoir's IPR obtained at previous timestep (Δt) as boundary condition for network simulator
Network model calculates solution for following timestep (Δt+1)
Pass rate and pressure solutions from network simulator to reservoir simulator
Reservoir simulator calculates solution for (Δt+1)

Figure 2.6: Schlumberger's *NetBalAct* coupling algorithm

NetBalAct can be configured for both explicit (*Periodic*) and partially implicit (*Iteratively Lagged*) simulations. If the logic in Figure 2.8 is executed once every time-step, then *Field Management* is executing its fully explicit coupled scheme. The *Iteratively Lagged* coupling setting incorporates balancing and convergence tests at the Newton-Raphson iteration level. Figure 2.9 illustrates the *Iteratively Lagged* coupling algorithm. When the *Iteratively Lagged* scheme is engaged, *Field Management* logic performs "coupled iterations". These coupled iterations involve the simultaneous evaluation of the network and reservoir simulators. During each coupled iteration, the reservoir conditions are held constant. This allows for the boundary conditions, as calculated by the network simulator, to balance and come into agreement with the reservoir simulator's solution. The coupled iteration improves mass balance convergence over time, yielding a more accurate production forecast as well as less non-physical oscillations. The user specifies the number of coupled iterations that *Field Management* will perform. If the convergence criteria are not met after the specified number of coupled iterations, *Field Management* will conduct uncoupled iterations. An uncoupled iteration holds the surface simulator solution constant, while allowing the reservoir to return to dynamic conditions. *Field Management* performs Newton-Raphson iterations within the reservoir simulator until the reservoir solutions converge with the fixed surface simulator solution. In this sense, *NetBalAct* solutions are controlled by the reservoir simulator. The property ID *WellUpdateNewtons* is set to the desired number of coupled iterations. *WellUpdateNewtons* is the coupled iterations keyword is specified within the *CouplingProperties* child node in the *fm_edits.ixf* file. [mengdi] showed that for a simple reservoir, three coupled iterations in the partially implicit coupled system yields results nearly equivalent to those of a fully implicit model created in MRST. In our simulations, we specified one coupled iteration per time-step. Therefore, we limited our system to one iteration of balancing. As prior work had shown that as few as three coupled iterations mitigated oscillations, we wanted to ensure that our base case coupled simulation experienced oscillations. Once we defined a base case that experienced oscillations, we could then test the effectiveness of PID controlled time-stepping in the simulation. Therefore setting *WellUpdateNewtons* equal to one was an experimental control that allowed us to confirm if our automatic time-stepping system did indeed reduce oscillation. In addition, we specified report times in the *fm_edits.ixf* file that corresponded to our automatic time-steps. Doing so ensured that our simulation would perform the coupling action in accordance with our PID controller output time-steps.

As noted, Schlumberger's *Field Management* software does not offer a fully implicit coupled simulation. However, its use of PIPESIM and INTERSECT allows for great flexibility in quickly designing production systems.

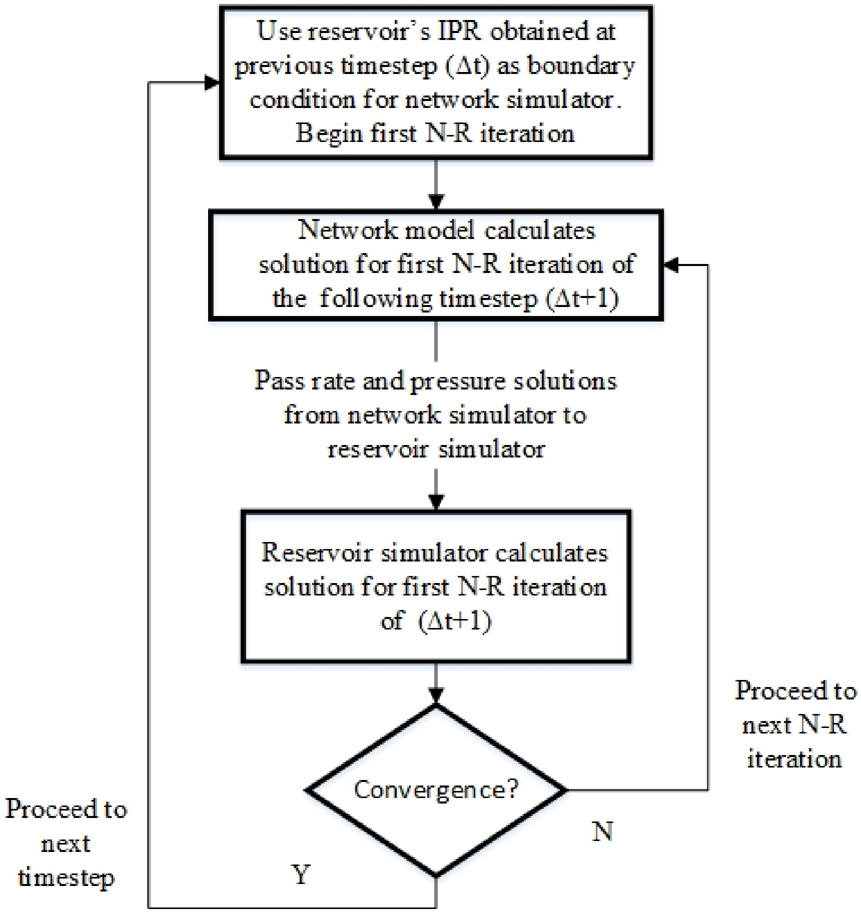
Use reservoir's IPR obtained at previous timestep (Δt) as boundary condition for network simulator. Begin first N-R iteration
Network model calculates solution for first N-R iteration of the following timestep (Δt+1)
Pass rate and pressure solutions from network simulator to reservoir simulator
Reservoir simulator calculates solution for first N-R iteration of (Δt+1)
Convergence?
Proceed to next N-R iteration
N
Proceed to next timestep
Y

Figure 2.7: Schlumberger's *Iteratively Lagged* coupling algorithm

In summary, chapter two provided theoretical background on surface and subsurface simulation. Chapter two also discussed in detail both general coupling algorithms and coupling algorithms used in this study. Chapter three will describe control theory, and introduce the controller used in this study. In addition, the complete coupling scheme including the PID controller will be described.

CHAPTER III

PID CONTROLLED TIME-STEPPING

In this chapter, general control theory and formulation are discussed. In addition, the error oriented PID controller used in our experiment is derived. The use of PID control for adaptive time stepping applications is discussed. The full adaptive time-stepping algorithm (PID control within the *Iteratively Lagged* framework) will be described. As a note, general controllers are derived using the dynamic equations of physical processes. The controller used in this study is was not derived using equations of physical systems. It was derived assuming a power relationship between error and time-step size. That is, it was derived to reduce error oscillations (eliminate oscillatory behavior), and not eliminate steady-state error.

3.1 Introduction to control

Controllers have been used to control dynamic systems for nearly a century. Early controllers were mechanical devices. Such simple controllers regulated steam engines, furnace temperatures, and other industrial processes. Later, advanced controls were implemented in processes such as chemical synthesis, automated manufacturing, and electricity generation. Today, many controllers are digitally based. In general, controllers are designed for specified, dynamic systems. The dynamic systems are typically described as the “plant”, i.e. the physical processes. Example processes for which controller may be derived include fluid flow, heat transfer, and competing chemical reactions inside a reactor. A controller will typically measure one variable in the dynamic system, and force a change in another variable to achieve a desired operating condition in the plant. There are two general types of controllers: open-loop and closed-loop. Figures 3.1 and 3.2 illustrate open loop and closed loop control, respectively. In open-loop control, a user specified, optimal operating signal (setpoint) is fed to the controller. The controller then sends a signal to the plant, which adjusts its operating conditions. Should a disturbance change plant conditions, there is no means for the controller to automatically adjust. The closed-loop, or feedback control system facilitates automatic adjustment to unfavorable disturbances in operating conditions. Our simulation utilized closed-loop control.

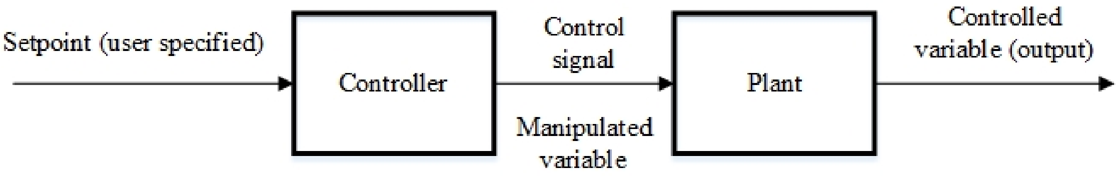

Figure 3.1: Open-loop control system

Closed-loop feedback control measures error between the setpoint (optimal condition) and controlled output (actual output). Error is input into the controller.

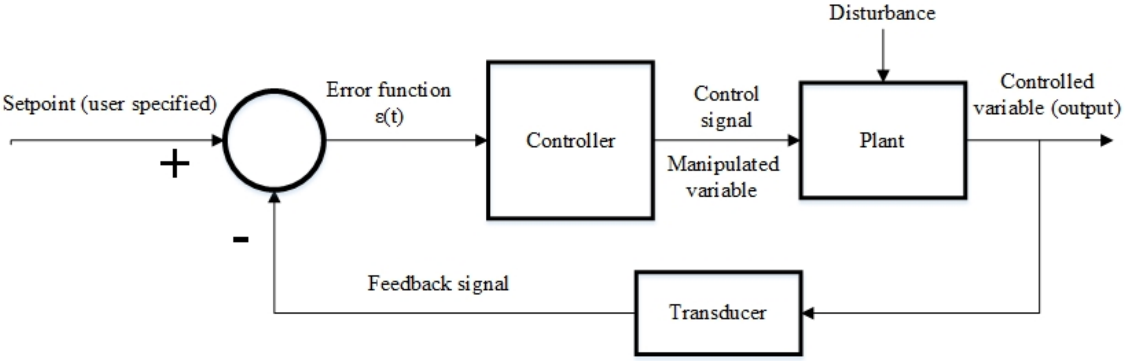

Figure 3.2: Closed-loop control system

The goal of the closed-loop system is to ensure that a specific plant output, the controlled variable, remains close to the user defined setpoint across time. Ideally, the setpoint and controlled variable should be equal. However, this is not possible in real dynamic systems. Controllers typically behave by forcing the manipulated variable closer to the setpoint across time in an oscillatory manner.

3.2 Dynamic system modeling

Figure 3.2 illustrates how discrete dynamic systems are described in partitioned blocks. In control theory, transfer functions are used to describe the dynamic properties for each individual block. The dynamic equations of each block are initially defined in the time domain. Laplace transformations are typically employed to convert all dynamic equations from the time domain to the s or Laplace domain, in order to yield the transfer function. Using Laplace transforms requires that the differential equations must be linear/linearized. Figure 3.3 illustrates the time domain input *x(t)* and time domain output signal *y(t)* of the dynamic system or controller block *h(t)*.

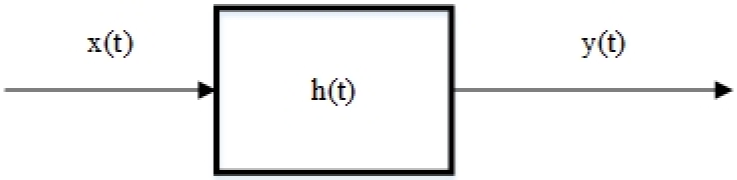

Figure 3.3: Control block h in time domain

Figure 3.4 illustrates the same system transformed, via Laplace transforms, to the s domain. Because convolution in the time domain becomes multiplication in the s domain, we can easily determine an unknown output, given we know the input as well as the dynamic system.

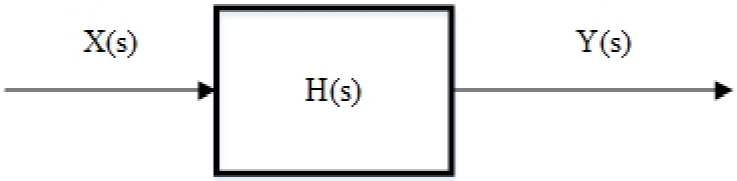

Figure 3.4: Control block H in s domain

In the s domain, the transfer function of *H* is defined as:

$$H(s) = \frac{Y(s)}{X(s)} \qquad (3.1)$$

The output of the system can be then solved:

$$Y(s) = X(s)H(s) \qquad (3.2)$$

The purpose of the transfer function is to enable differential equations to be solved with simple algebraic manipulation. Controllers are derived using transfer functions. There are many types of controllers, however this study uses a PID controller.

3.3 PID control

PID control is one of the most widely used controls across all industries. Early PID control was created after engineers observed that naval helmsmen steer ships based upon current error, past error, and rate of change of error. Early PID control was used to steer large vessels. Modern PID control is implemented in aircraft altitude control, distributed control systems in manufacturing, and other complex processes. The PID controller is comprised of three sub-controllers: proportional, integral and derivative control. Each of these components provide respective, characteristic behavior to the PID controller's response. The proportional component responds to instantaneous changes in the controlled (measured) variable. For example, when a controller responds to a brief, transient change, the response is primarily due to proportional action. The integral control responds to cumulative error. This control measures error over time, and will reduce steady-state error. The derivative control compensates for future error by taking action based upon the instantaneous rate of change in the plant output. The time domain PID controller can be expressed as [26]:

$$u(t) = K_p * \varepsilon(t) + K_i * \int \varepsilon(t) + K_{d*} \frac{\partial \varepsilon}{\partial t} \qquad (3.3)$$

In addition, the transfer function, *G*, of a general PID controller in the Laplace domain is illustrated with the following equation [26]:

$$G_{PID}(s) = \frac{K_i + K_p s + K_d s^2}{s^2} \qquad (3.4)$$

where K_p is the proportional coefficient or gain, K_i is the integral gain, and K_d is the derivative gain. These gains are modified, or "tuned", to provide a desired controller response. The desired controller response will depend on the characteristics of the system being controlled, and the objective of the user. There are tuning algorithms that utilize various parameters and heuristics. In this study, the PID controller is manually tuned. As a note, the majority of existing tuning algorithms were not created for adaptive time-stepping applications.

3.4 Experimental PID controller derivation

Figure 3.5 below illustrates the closed-loop system in the Laplace domain. This general controller scheme was selected for use in our experiments. [1] used the same closed loop system to perform adaptive time-stepping in reservoir simulation.

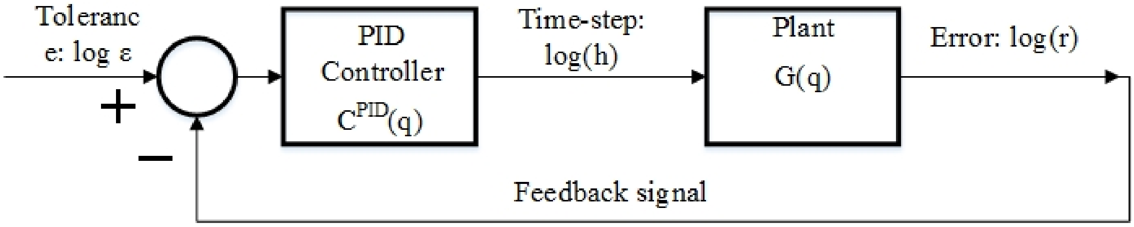

Figure 3.5: Block diagram of experimental time-stepping closed loop system

In Figure 3.5, h is the time-step controller output, ε is the user specified tolerance, r is the error, G(q) is the transfer function of the plant, and C^{PID} is the PID controller transfer function. Error is a plant output and is the controlled variable. [14] derived the following controller to controller to reduce solution oscillations in automatic step-size control in the integrations of ODE's. The controller is unique in that it is not a linear combination of the proportional, integral and derivative terms. [14] observed that for the nonlinear system (ODEs), the error at each time-step is dependent on the step-size asymptotically. The controller was not derived to reduce error to zero- it was derived to reduce oscillations in error, and thus ensure convergence and timely solutions when integrating ODEs. The error tolerance was treated such that error should remain stable (non-oscillatory) at a value less than the tolerance. We selected this same controller because we wanted to control the rate of change of error (to reduce oscillations), and not error itself. We also believed that coupling error is similarly proportional to the step-size as in [14]. [38, 1] chose the controller derived in [14]. The following equations highlight the use of the backward shift operator. For the full derivation of the PID controller, please refer to [14].

The discrete time domain PID controller transfer function is obtained by implementing the backward shift operator as utilized by [8]:

$$q^{-1} * u_n = u_{(n-1)} \qquad (3.5)$$

where u_n is the output of the PID controller at time-step *n*, and $u_{(n-1)}$ is the output of the PID controller at time-step *n-1*. In addition, of backward shift operator can be substituted into the following equation:

$$u_n - u_{(n-1)} = u_n - q^{-1}u_n \quad (3.6)$$

The difference between consecutive iterations can be further simplified:

$$u_n - u_{(n-1)} = (1 - q^{-1}) * u_n \qquad (3.7)$$

$$u_n - u_{(n-1)} = \frac{q-1}{q} u_n \qquad (3.8)$$

$$u_n - u_{(n-1)} = \nabla f \qquad (3.9)$$

where ∇ is the backward difference operator. The backward difference operator is analogous to the derivative operator in a continuous function. Likewise, the summation operator $\frac{q}{q-1}$ is analogous to integral operators defined for continuous functions. Both operators are utilized to realize the transfer function of the PID controller in the time domain:

$$u_n = e_n \left[K_p + \frac{q-1}{q} * K_d + \frac{q}{q-1} * K_i\right] \qquad (3.10)$$

where e_n is the input error function. The error function is the difference between the user defined tolerance and the output plant error, as seen in Figure 3.4:

$$e_n = \log(\varepsilon) - \log(r_n) \qquad (3.11)$$

Applying the backshift operator:

$$e_n = q^{-1} \, [\log(\varepsilon) - \log(r_n)] \qquad (3.12)$$

Likewise, Figure 3.4 shows that the PID output in (3.10) can be substituted for:

$$u_n = \log(h_n) \quad (3.13)$$

Substituting equations 3.12-3.13 into equation 3.10 yields:

$$\log(h_n) = q^{-1}\left[K_p + \frac{q-1}{q} * K_d + \frac{q}{q-1} * K_i\right] * [\log(\varepsilon) - \log(r_n)] \quad (3.14)$$

The PID transfer function is:

$$C^{PID}(q) = q^{-1}\left(K_p + \frac{q-1}{q} * K_d + \frac{q}{q-1} * K_i\right) \quad (3.15)$$

Further substitution yields:

$$\log(h_n) = C^{PID}(q)[\log(\varepsilon) - \log(r_n)] \quad (3.16)$$

By substituting the backward shift operator with ∇, and determining the difference in equation 3.14:

$$\log(h_{n+1}) - \log(h_n) = K_I(\log(\varepsilon) - \log(r_n)) - K_p(\log(r_n) - \log(r_{n-1})) - K_D(\log(r_n) - 2\log(r_{n-1}) + \log(r_{n-2})) \quad (3.17)$$

Rearranging to simplify log operators:

$$\log\frac{h_{n+1}}{h_n} = K_I \log\frac{\varepsilon}{r_n} - K_P \log\frac{r_n}{r_{n-1}} - \ K_D \log\frac{r_n * r_{n-2}}{r_{n-1}^2} \quad (3.18)$$

After applying the exponential to 3.17:

$$\frac{h_{n+1}}{h_n} = \left(\frac{\varepsilon}{r_n}\right) K_I * \left(\frac{r_{n-1}}{r_n}\right)^{K_P} * \left(\frac{r_{n-1}^2}{r_n * r_{n-2}}\right)^{K_D} \qquad (3.19)$$

finally, our adaptive time-stepping PID controller can be fully described in the time domain:

$$h_{n+1} = h_n * \left(\frac{\varepsilon}{r_n}\right) K_I * \left(\frac{r_{n-1}}{r_n}\right)^{K_P} * \left(\frac{r_{n-1}^2}{r_n * r_{n-2}}\right)^{K_D} \qquad (3.20)$$

Equation 3.19 is the general controller used in our experiments. This controller and its derivation is fully described in [15]. During simulation, this controller responds to error in the current time-step solution, as well as error in the previous two time-steps. The next sub-chapter describes our definition of error in our experiments.

One important aspect of controller design is the stability of the controller, which is not discussed in this thesis. Additional discussion on adaptive time-stepping and stability can be found in [35].

3.5 Error definition for coupled Systems

We chose the INTERSECT property ID *NCEP* as our error input. *NCEP* is Schlumberger's acronym for Network Coupled Error Pressure, and is defined as:

$$r_n = NCEP = \frac{\left|P_{IX,n}^{coupling\ point} - P_{PIPESIM,n}^{coupling\ point}\right|}{P_{IX,n}^{coupling\ point}} \quad (3.21)$$

where $P_{IX,n}^{coupling\ point}$ is the pressure calculated at the coupling point by INTERSECT for time-step n, and $P_{PIPESIM,n}^{coupling\ point}$ is the pressure calculated at the coupling point by PIPESIM at time-step n. In effect, *NCEP* measures the difference between IX's and PIPESIM's solution for a shared boundary condition. The coupling point is the shared boundary condition that is passed from one the surface simulator to the subsurface simulator during the *Iteratively Lagged* coupling iterations. We defined the coupling point as the tubing head pressure.

In our experiments, we selected an error tolerance of 1. Error tolerance impacts the response of the integral control. Our PID controller simplified to:

$$h_{n+1} = h_n * \left(\frac{1}{r_n}\right) K_I * \left(\frac{r_{n-1}}{r_n}\right)^{K_P} * \left(\frac{r_{n-1}^2}{r_n * r_{n-2}}\right)^{K_D} \quad (3.22)$$

Coupling error at each time-step, *NCEP* or equivalently, r_n, was specified as an output in the *RSM* file. The *RSM* file is a summary results text file generated by INTERSECT/*Field Management* at the end of each simulation. NCEP can be specified as an output for individual producers. INTERSECT does not have the capability to determine coupling error for injectors. The user specifies the desired quantities to be output in the file. The next sub-chapter describes the entire adaptive time stepping algorithm.

3.6 Error control adaptive time-stepping algorithm

The complete error control, adaptive time-stepping algorithm is illustrated in Figure 3.6 below:

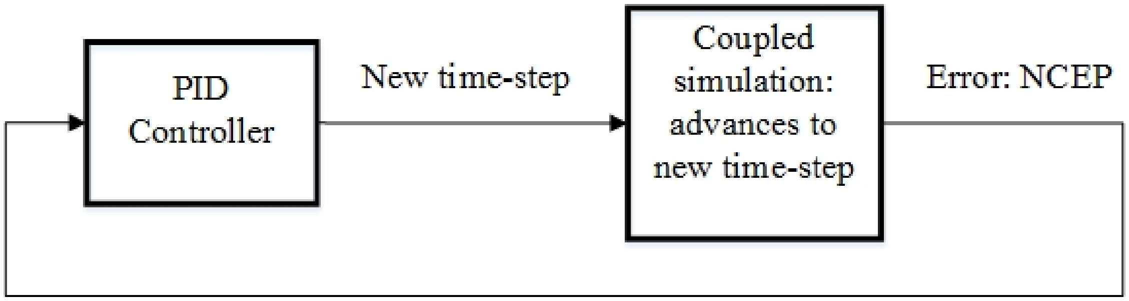

Figure 3.6: Adaptive time-stepping algorithm

In our experiments, the *Iteratively Lagged* coupled simulation, with *WellUpdateNewtons* equal to one, was treated as our “plant”. The controlled variable was *NCEP*, and the manipulated variable was the time-step. The integration of the PID controller and the coupled simulation was performed manually. The simulation was initialized with three small time-steps programmed in the *IXF* files. These first three time-steps provided the PID controller with the initial three values of error, as equation 3.21 illustrates the controller requires error inputs from the current and previous two time steps. These three initial error values were fed into the PID controller in MATLAB, which output the new time-step. This new time-step was then programmed into the *IXF* files, and the simulation executed to the end of the updated simulation time. The updated error was then passed to the PID controller, and the process repeated.

In summary, the PID controller received error outputs from the coupled simulation and calculated optimized time-steps designed to reduce coupling error. Our hypothesis was that by reducing coupling error, we would reduce non-physical oscillations in the coupled system pressure and rate solutions. We hypothesized that the use of a PID controller [14] to automatically select the time-step would reduce coupling error and non-physical oscillations. This discussion concludes the derivation and operation of the controller and adaptive time-stepping algorithm.

The next chapter discusses the base case simulation as well as our experiments and results.

CHAPTER IV

EXPERIMENTAL DESCRIPTION AND RESULTS

The following sub chapters describe both the underlying reservoir simulation and underlying network simulation used in coupling, as well as the experimental organization and results.

4.1 Base case description- INTERSECT reservoir description

All simulations featured the same reservoir, RES1. RES1 featured a structured, Cartesian grid. RES1 also featured uniform permeability in the i, j, and k directions, respectively. See Table 4.1 below for additional reservoir and grid properties. In a sense, this reservoir is a “shoe box”. However, testing the PID controller in this system allowed us to execute a proof of concept within a widespread commercial simulator. The testing of a more complex reservoir can easily be accomplished within the framework established in this body of work.

Property	RES1
$N_x : N_y : N_z$	10:10:12
Δx: Δy: Δz	250x250x10
Permeability (mD) (i, j, k)	100, 100, 10
Porosity (%)	30
Datum (ft)	9150
Datum pressure (PSI)	4082

Initial gas oil contact (ft)	9150
Water oil contact (ft)	9350
Bubble point at datum (PSI)	4082

Table 4.1: Description of reservoir model used in simulation

Figure 4.1 below illustrates the initial pressure distribution in RES1. The slight gradient in the bottom six grid blocks is solely due to gravitational effects.

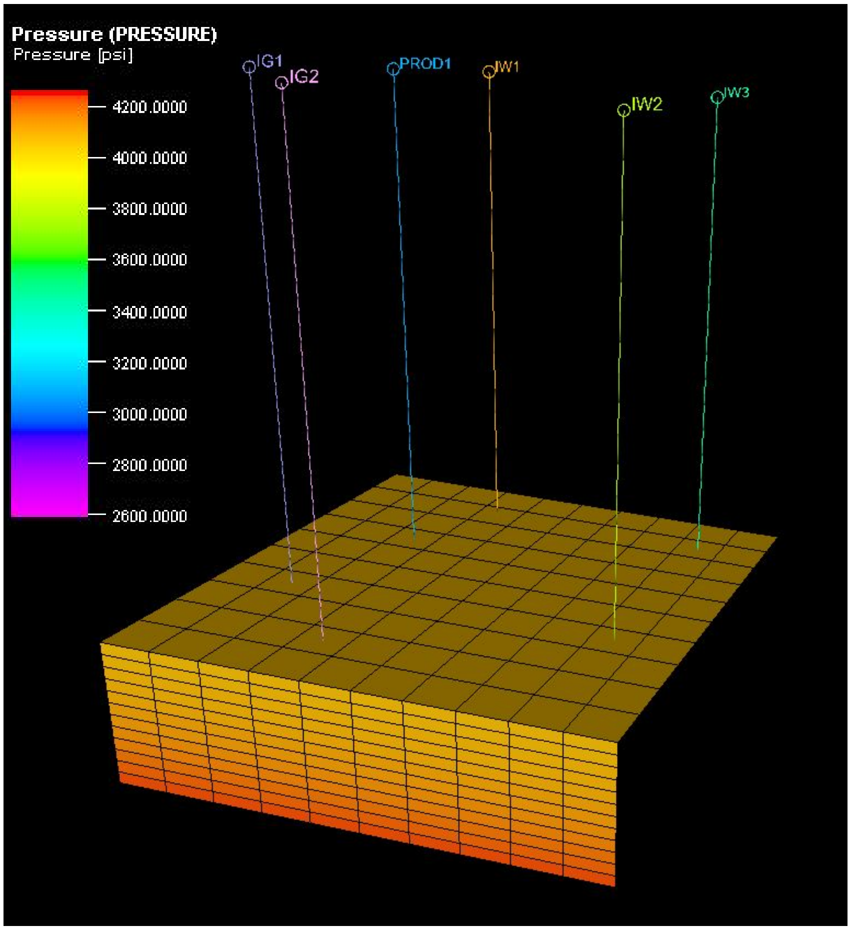

Figure 4.1: RES1 initial pressure distribution

Two spatial well configurations were tested. Scenario 1 features a single producer, PROD1. Scenario 2 features two producers, PROD 1 and PROD 2. Both scenario 1 and scenario 2 include three water injection wells (IW1, IW2, IW3) and two gas injection wells (IG1, IG2). Below, figure 4.2 illustrates the spatial configuration of the wellbores in the x-y plane. During scenario 1 simulation, PROD 2 was effectively plugged.

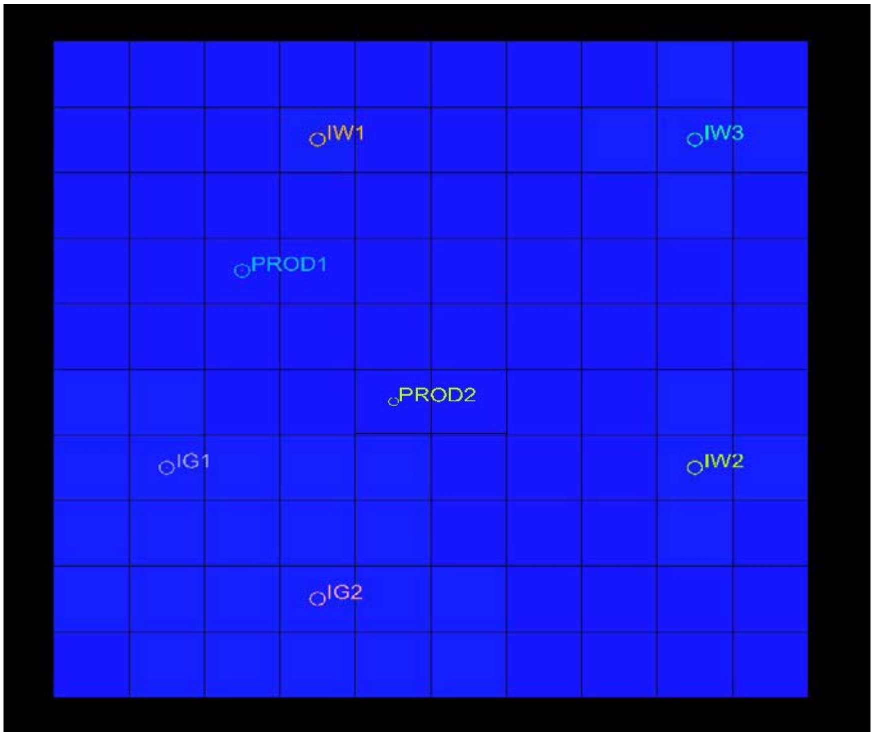

Figure 4.2: Scenario 2 wellbore configuration in INTERSECT simulation

Two separate production/rate constraints were tested. PROD1 in scenario 1 was constrained to 10,000 STB/Day, and bottomhole pressure was left unconstrained. Bottomhole pressure in PROD1 and PROD2 was constrained to 2,500 PSIA when the

Scenario 2 well configuration was simulated. Table 4.2 illustrates the injection rates of the water and gas injection wells:

INTERSECT setting	IW1	IW1	IW3	IG1	IG2
Injection Rate (BWPD/MSCFD)	5000	5000	5000	5000	5000
BHP (PSI)	4800	4800	4800	4800	4800

Table 4.2: INTERSECT injection well settings

All wellbores were modeled in INTERSECT. INTERSECT treated the wellbore as additional gridblocks. For each well-to-cell connection, k, INTERSECT calculated the following IPR:

$$q_a^k = T^k \lambda_a^j \left(P^j + \Delta P^{kj} - P^w - \Delta P^{kW}\right) \qquad (4.1)$$

where T^k is the well-to-cell connection transmissibility for connection k, and

λ_a^j is the ratio of relative permeability of the phase *a* divided by its viscosity for grid cell j. P^j is the cell's pressure used in drawdown calculations, $\Delta P^{kj}$is the hydrostatic pressure difference within the cell between the cell center depth and the connection center depth, $P^w$is the pressure at the well's segment node, and ΔP^{kW} is the hydrostatic pressure difference within the well between the connection center depth and the depth of the well's segment node. Refer back to chapter two for numerical aspects of reservoir simulation.

4.2 Base case description- PIPESIM network description

PIPESIM was used to model pressure drop across the surface network. The surface network included two well point sources (PROD 1 and PROD 2), two flow lines, and a major trunk line terminating in a sink (GATHER). Figure 4.3 illustrates the model. All piping was 9'' ID, with 0.5'' wall thickness and 0.001 roughness. PIPESIM's blackoil model was used (to match INTERSECT's settings). The source streams were specified at 120 degrees Fahrenheit, and the ambient pipeline at 60 degrees Fahrenheit. The sink, GATHER, operated at a constant 300 PSIA, suction.

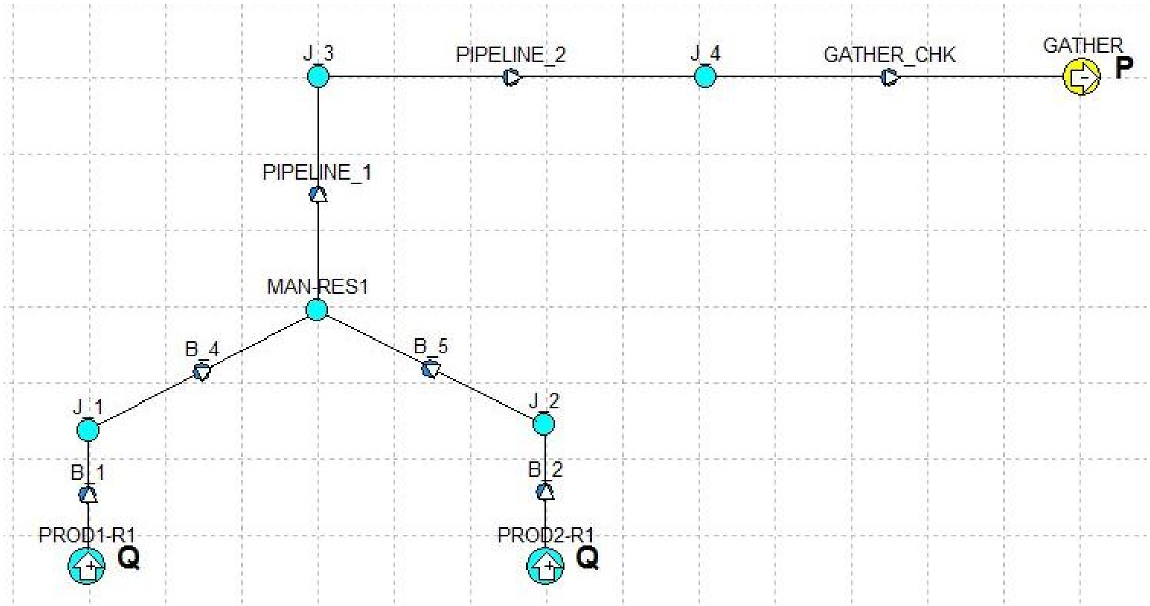

Figure 4.3: PIPESIM surface network used in all experiments

4.3 Base case description- partially implicit coupling

Prior to utilizing the PID controlled adaptive time-stepping algorithm as outlined in figure 3.5, PIPESIM and INTERSECT were tested using *Field Management*. This base case was performed to supply an unstable coupling benchmark for initial PID tests. This base case was performed using the same relaxed settings outlined in chapter two. Again, the purpose of testing relaxed settings in both the base case and the PID case was to ensure an "apples to apples" comparison. The "apples to apples" comparison allowed the effect of PID control on oscillation reduction to be scrutinized. Our PID controlled results could then be compared to the base case and be judged with clarity. We executed one base case simulation for every PID controlled simulation. Each PID controlled simulation and base case simulation pair terminated at the same date. Having normalized base cases for each PID experiment allowed us to create standardized procedure to consider the effectiveness of each PID controlled experiment. For the base case, and subsequent experiments, the coupling error, coupling pressure, bottom-hole pressure, production rate, and cumulative production data was collected and graphically analyzed. The coupling pressure was the pressure value collected at the coupling point. In our experiments, the coupling pressure was the tubing head pressure. This was done for practical reasons. Selection of the tubing head as the coupling point allows for intervention in simulation, to reflect possible optimization or operational requirements. The base case is interchangeably referred to as the IX/FM base case or simulation. IX and FM is Schlumberger's acronym for INTERSECT and *Field Management*, respectively. This section will introduce and analyze the behavior of the base case. The aim of this section is to provide a basic understanding of how Schlumberger's software handles coupling. The analyst may use this understanding to gauge the effectiveness of our PID control used in Schlumberger's software.

The first base case example, below in Figure 4.4, highlights coupling error over time. Coupling error is defined in INTERSECT as:

$$Coupling\ Error = \frac{Tubing\ Head\ Pressure_{INTERSECT} - Tubing\ Head\ Pressure_{PIPESIM}}{Tubing\ Head\ Pressure_{INTERSECT}} \tag{4.2}$$

where the tubing head pressure is the coupling pressure. If the bottom-hole pressure is selected as the coupling point, then equation 4.2 will change to reflect error at the bottom-hole. The coupling error measures the departure of the surface and sub-surface simulators from one another. In Figure 4.4, we notice that the error is oscillatory. The error is also bounded by 200%. Schlumberger's *Field Management* outputs 200% coupling error whenever the reservoir cannot deliver the flowrate the surface network calculated for the time-step. In a representative surface-subsurface coupled system, this is analogous to a scenario in which the VFP and IPR share no solution. From a physical point of view, this means that there is no flow from the reservoir to the surface. However, the cause is not physical, it is a result of coupled simulation. In our case, from a simulation point of view, the independent surface and subsurface solutions have diverged enough that there are no shared set of solutions between the VFP and IPR.

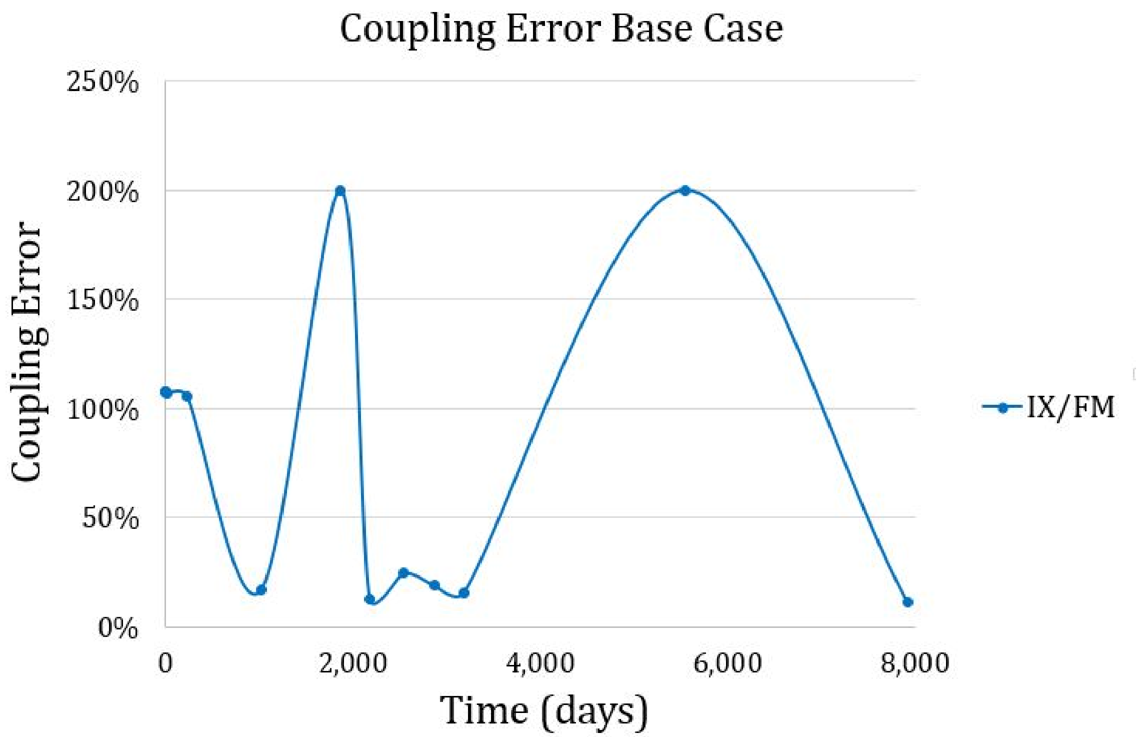

Figure 4.4: Base case error in coupled simulation

The tubing head pressure (coupling pressure) is also oscillatory. Figure 4.5 illustrates the base case tubing head pressure. In fact, the tubing head pressure is approximately -1,200 PSIA at approximately 5,500 days. This negative pressure is physically impossible, and is a result of the IPR (INTERSECT) not being able to deliver the flow rates calculated by the VFP (PIPESIM). This is evidence of unstable solutions present in the simulation. In INTERSECT, the reservoir controls the reported solution, so should the coupled solution VFP and IPR not converge, a modified reservoir solution will be reported.

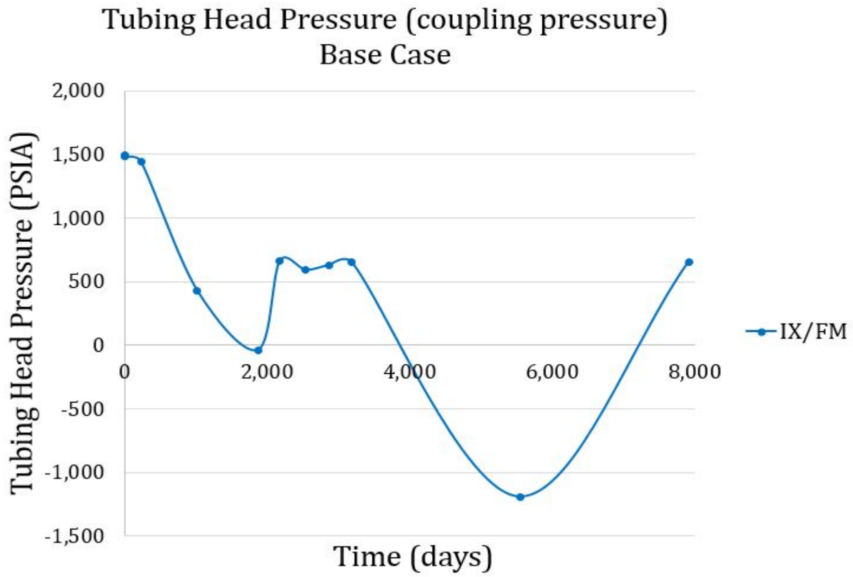

Figure 4.5: Base case tubing head pressure in coupled simulation

The bottom-hole flowing pressure is also oscillatory. Figure 4.6 illustrates the bottom-hole flowing pressure of the base case.

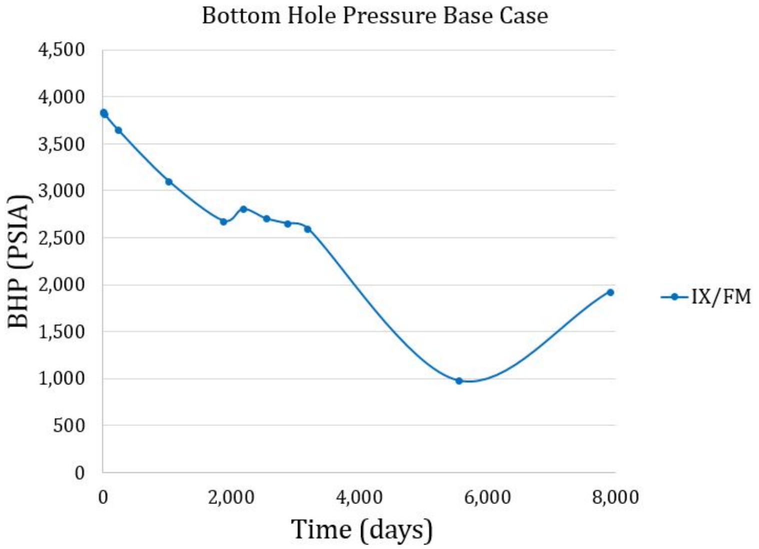

Figure 4.6: Base case bottom-hole pressure in coupled simulation

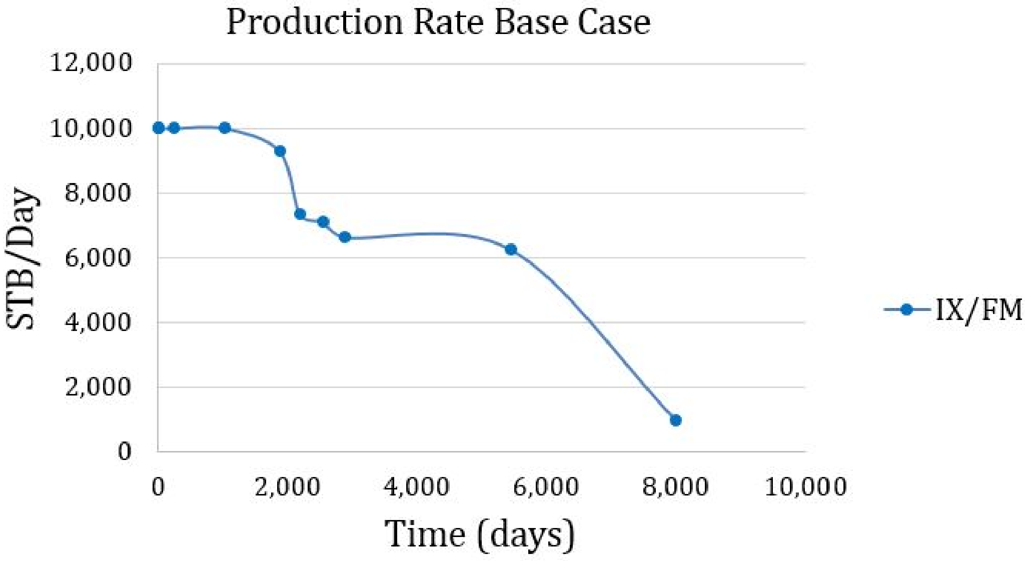

Figure 4.7: Base case production rate in coupled simulation

Figure 4.7 above illustrates the production rate profile for the base case coupled simulation. All four base case output plots show oscillations. In addition, the base case tubing head pressure solutions include physically impossible values. Therefore, we note the relatively high degree of non-physical oscillation in the base case coupled solution.

4.4 Initial test: heat transfer gain coefficients

We first tested a set of gains that were empirically determined by [25] in a PID controlled adaptive time-stepping heat transfer simulation. Relaxing *TimeStepSizeControls* constraints allowed us to implement the PID controller calculated time-steps. As stated earlier, the base case also has the same relaxed constraints in this experiment. Our initial results are illustrated in figure 4.8. Figure 4.8 illustrates the error across time determined by the PID controller (on a per time-step basis), overlaid with the base case error profile. It is apparent that the PID controller quickly reduces error by taking many small time-steps, and gradually increases the size of the time-step. In this experiment, and all other experiments, tolerance was specified at 1. According to the original derivation [14], setting the tolerance to 1 should keep error bounded by 1. Figure 4.8 shows error stabilize at about 45%. Therefore, observed error control agrees with the intended theoretical error control.

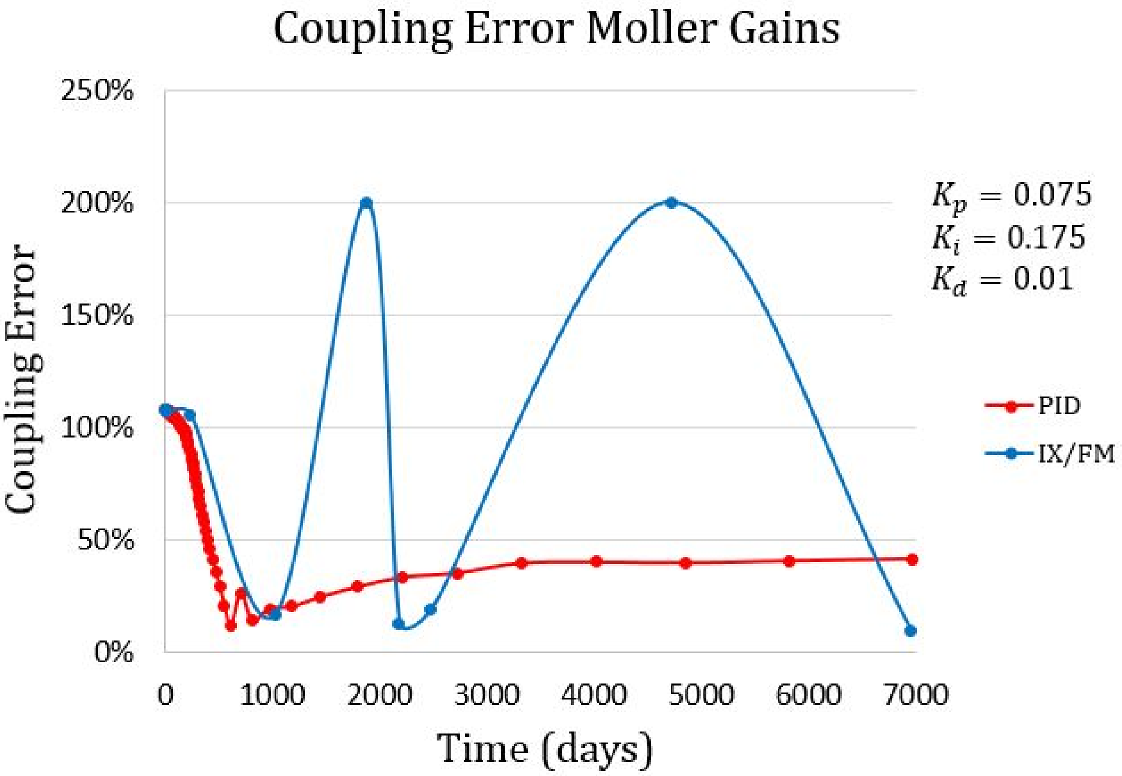

Figure 4.8: Coupling error for first PID test using [25] gains

Our first experiment motivated us to conduct additional experiments, as well as manually tune the controller gains to understand how gain selection influences the error control. In the following experiments, the heat transfer experiment [25] gains were tested. However, we varied one gain value in each experiment, while keeping the other gains constant.

4.5 Manual tuning: $\boldsymbol{K_p = 0.1}$

In the following experiment, K_p was specified as 0.1, while K_i and K_d were held constant at [25] gain values. The relaxed constraints were maintained in both the PID and base case simulations. Figure 4.9 below illustrates coupling error for this experiment. Again, the PID controller drastically reduces error oscillation as compared to the IX/FM base case. Changing K_p to 0.1 increases the initial controller response. As compared to Figure 4.8, error is initially decreased to a smaller value. However, changing K_p to 0.1 slightly increases the oscillatory behavior of the PID controller. K_p becomes unstable between the values of 0.5 and 1.0. Later, an unstable controller K_p= 1.0 is introduced and discussed. It is easy to observe that small time-steps are associated with error reduction with the PID controller. As a reference, the base case executed 10 time-steps, while the PID controller executed over 60. Typically, oscillations occur in the base case following a succession of small time-steps. This is likely due to the underlying time-step selection algorithm. During a succession of small time-steps, saturation and pressure changes within individual cells are likely small. In addition, the time truncation error is likely small. The base case's relaxed settings then dictates a disproportionally large time-step, resulting in coupling error. The last time-step in the base case shows small coupling error. This is likely due to the stabilization of the reservoir.

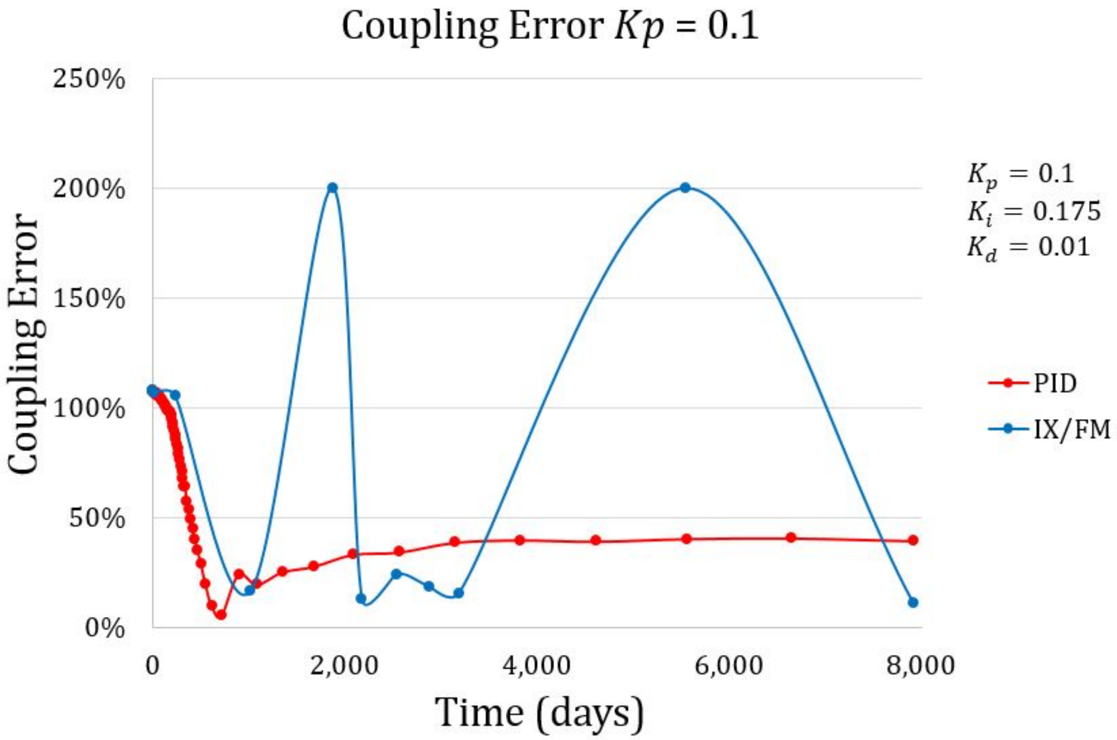

Figure 4.9: Coupling error for manual gain test K_p = 0.1

Figure 4.10 below illustrates the tubing head pressure (coupling pressure) as calculated using PID control with K_p was specified as 0.1. We see that all non-physical behavior is eliminated with the PID controller selecting the time-step. The PID controller selects time-steps that yield a smooth pressure decline as expected.

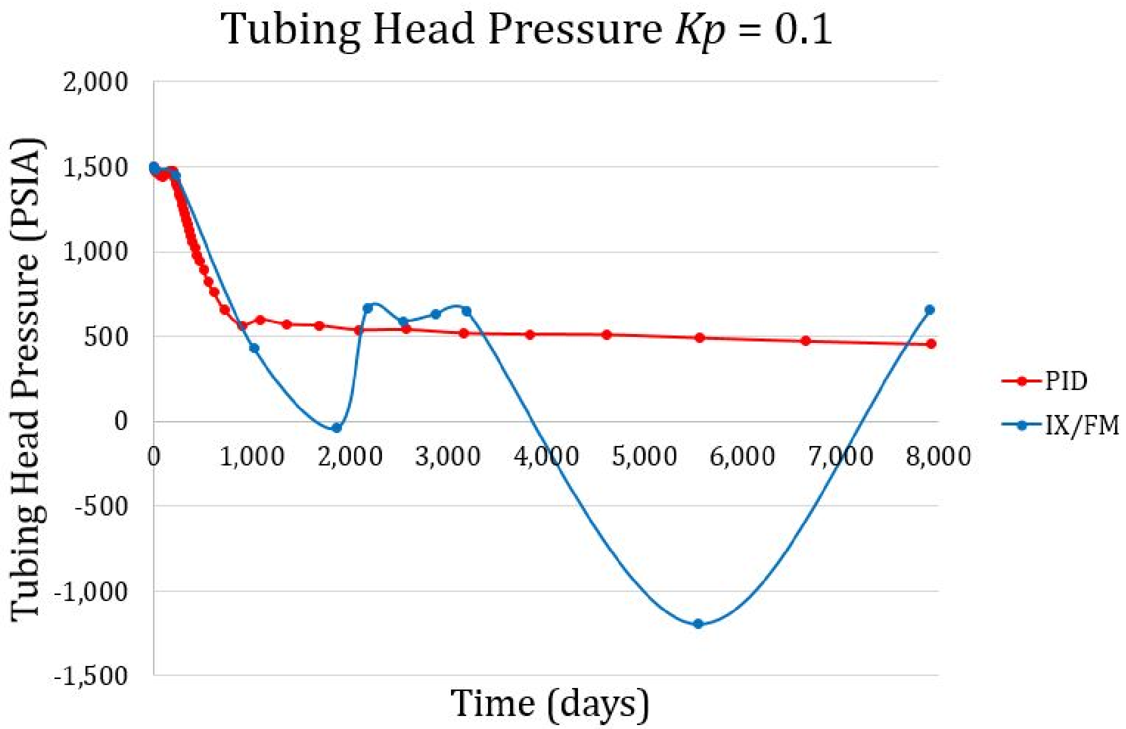

Figure 4.10: Tubing head pressure for manual gain test K_p = 0.1

It is of interest to note that the base case IX/FM simulation approximates the PID solution between 2,000 and 3,000 days. Figure 4.11 illustrates the bottom-hole pressure as calculated using PID control with K_p was specified as 0.1. Again, we see that all non-physical behavior is eliminated with the PID controller selecting the time-step. Again, we observe a smooth decline. We also observe that the base case IX/FM solution closely approximates the PID controlled solution. However, the bottom-hole pressure solution more closely approximates the base case solution than that of the tubing head pressure. This is likely due to the fact that the VFP is not a linear relationship, i.e. the bottom-hole flowing pressure and the tubing head pressure do not vary in a linear fashion, with

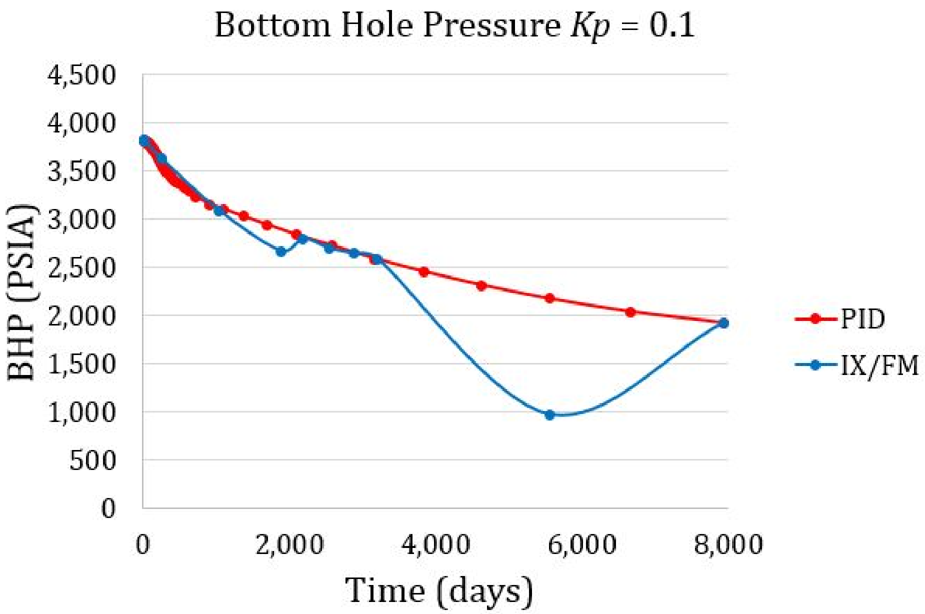

Figure 4.11: Bottom-hole pressure for manual gain test K_p = 0.1

respect to one-another. The next selected output for this PID controlled experiment is the production rate plot. This plot, Figure 4.12, is below.

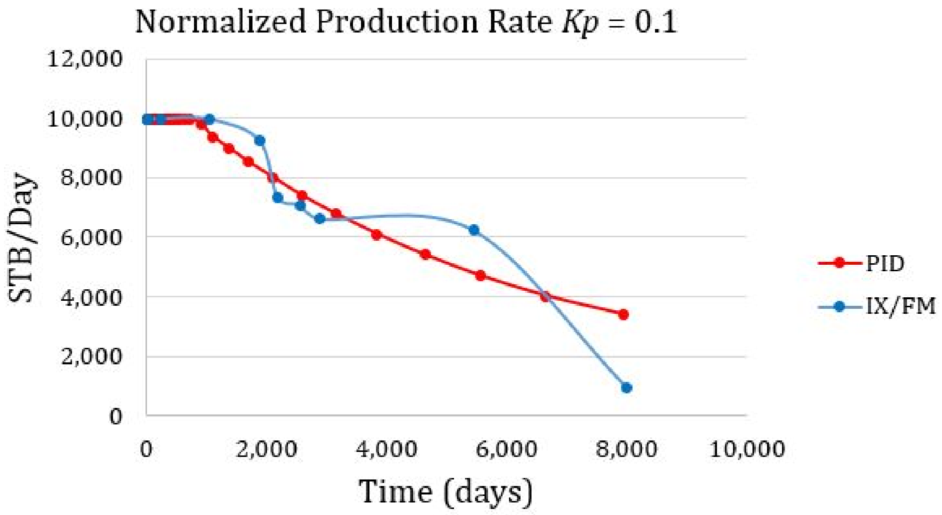

Figure 4.12: Normalized production rate for manual gain test K_p = 0.1

Again, we see the PID controller reduces oscillations. The PID controller yields a smooth, physical production rate decline curve as expected. Figure 4.13 below illustrates the cumulative production plot for this experiment.

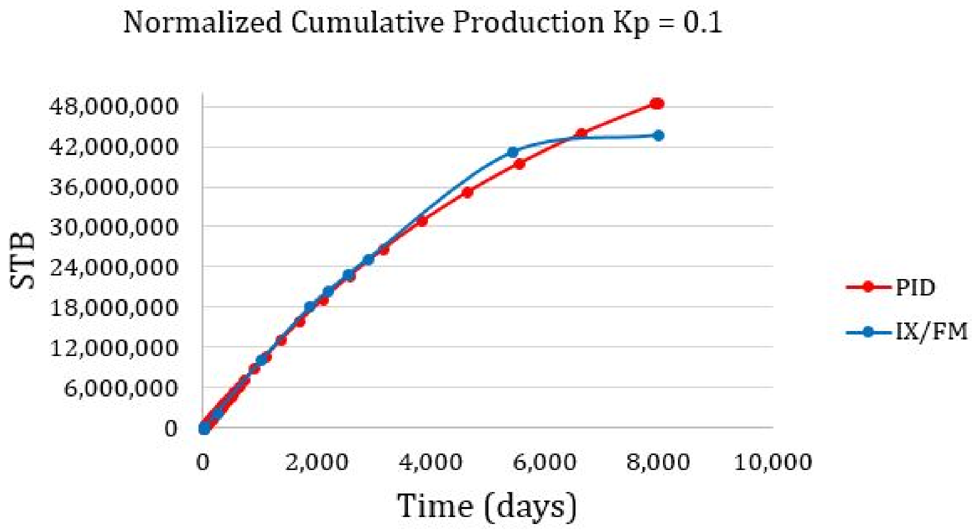

Figure 4.13: Normalized cumulative production for manual gain test K_p = 0.1

Cumulative production as calculated by the PID controlled time-step simulation was about 10% greater than that calculated by the base case simulation. It is apparent that the base case simulation does not predict accurate pressure and rates profiles, thus, the base case cumulative production is inaccurate. In addition, the last data point in the base case cumulative production plot is not significantly greater than the previous data point. This is because the last production rate calculated in the base case is unphysically small. It is apparent that the PID controller removes non-physical oscillations from an otherwise oscillatory simulation. Ten additional, parallel experiments were carried out. These tests were similar in that one gain was varied at a time, while the other two implemented were both gains from the heat transfer experiment [25]. The results are similar to the experiment described in this section. The next section will compare important parameters with respect to these simulations. Specifically, we will compare the simulation time required to execute all the various PID controller experiments.

4.6 Manually tuned PID computational costs

The five experiments illustrated below, in table 4.3, were all executed using the same relaxed *TimeStepSizingControls* as described in chapter two. The additional tests' results are similar with those presented in this section. The following data highlights the relative computational cost of PID control vs the relaxed base case.

$K_i = 0.175, \qquad K_d = 0.01$				
Experiment	K_p	PID CPU time (s)	IX/FM CPU time (s)	CPU Ratio
1	0.1	37.49	16.82	2.2
2	0.2	29.89	16.24	1.8
3	0.25	32.18	15.93	2.0
4	0.5	27.02	15.57	1.7
5	1	22.65	13.74	1.6

Table 4.3: Computational costs associated with coupled simulation

Results show that the PID controller takes up to twice as long to perform a simulation than the unstable base case. The PID controlled simulation was more computationally expensive because of the additional Newton-Raphson method. Table 4.4 below highlights the additional computational costs of PID control as compared to the unstable base case. The following data has been normalized in the experiments (common end date for all simulations). This allows for the direct comparison of simulation performance for different PID gains.

$K_i = 0.175, \quad K_d = 0.01$					
Experiment	K_p	PID coupled iterations	IX/FM coupled iterations	PID uncoupled iterations	IX/FM uncoupled iterations
1	0.1	760	509	2108	467
2	0.2	487	509	2081	467
3	0.25	567	509	2031	467
4	0.5	358	509	1953	467
5	1	312	509	1762	467

Table 4.4: Newton-Raphson iterations in coupled simulation

Coupled iterations are Newton-Raphson iterations for which the IPR is held constant while the surface simulator iterates to convergence within a time-step. Uncoupled iterations are Newton-Raphson iterations for which the VFP is held constant while the sub-surface iterates to convergence. The reservoir is considered the controlling simulation in *Field*

Management. That is, reservoir simulation convergence dictates overall coupled system convergence.

Tables 4.3 and 4.4 illustrate that as K_p is increased, computational costs decrease. That is, simulation time, coupled iterations and uncoupled iterations decrease as K_p increases. However, the controller becomes unstable for larger values of K_p. Figures 4.14 and 4.15 illustrate that the PID controller becomes oscillatory at K_p = 1. As a note, the controller became unstable for large values of K_i and K_d as well.

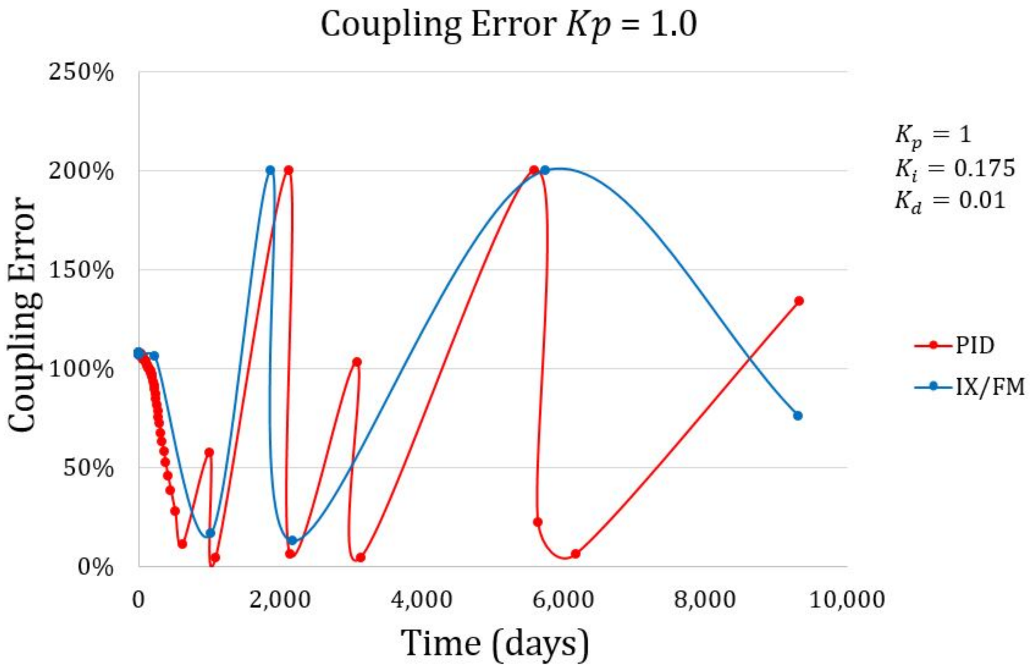

Figure 4.14: Coupling error for unstable controller, K_p = 1.0

We noticed that both the unstable controller and stable controller (K_p = 0.1) executed many early time-steps. However, the unstable controller still fails to reduce error. This leads us the conclusion that stability is not controlled solely by the number of time-steps executed. In fact, the size of the selected time-step is crucial. Figure 4.15 below illustrates tubing head pressure for $K_p = 1.0$. The tubing head pressure is oscillatory as expected.

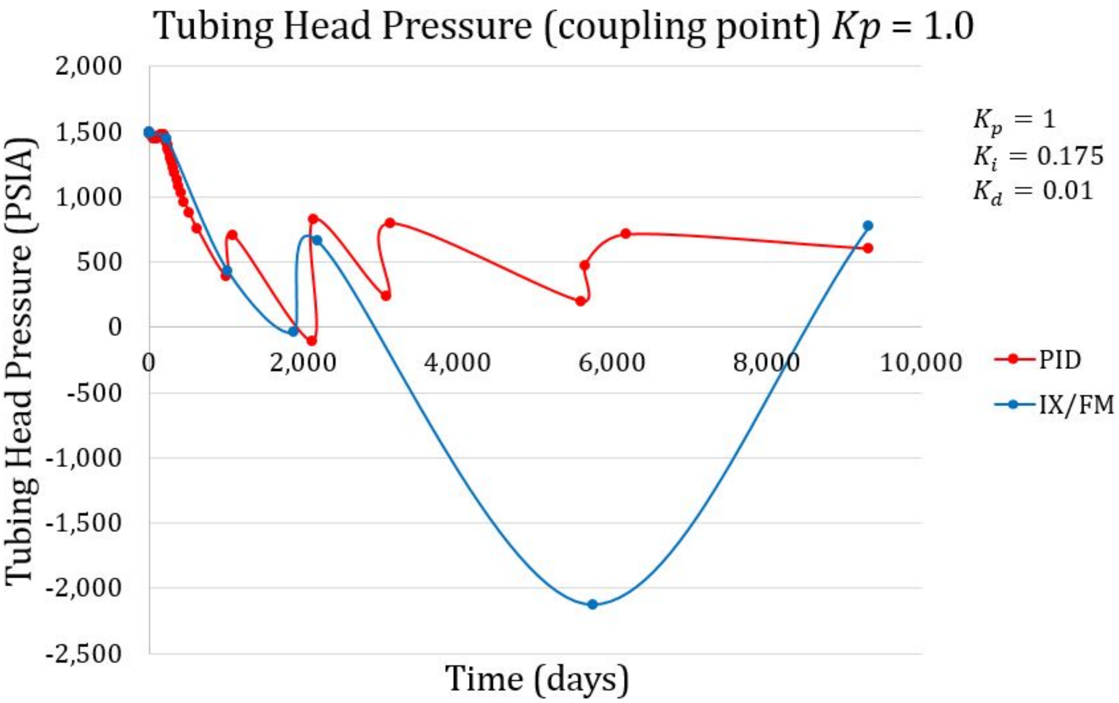

Figure 4.15: Tubing head pressure for unstable controller, K_p = 1.0

It is important to note that the base case IX/FM simulation was normalized such that the end date of simulation was equivalent to the end date of simulation for the PID case. The PID controlled simulation was ended arbitrarily between 7,000 and 8,000 days for each experiment. It is also important to consider that the relaxed base case has poor coupling time-stepping controls. Therefore, the computational cost comparison does not compare two adequate coupling methods. As noted, an additional experiment was conducted that compares PID performance to Schlumberger's coupling default settings, and it is discussed later. However, these results served to motivate the next section. The computational cost analysis serves to highlight the relative cost of PID control. The unstable controller illustrates that the size, and not just number, of time-steps are critical. Both computational cost and unstable controller discussions show that the PID controller gains must be optimized to reduce computational costs as well as ensure stability during simulation. The next section highlights a scheduled gain controller that was optimized to reduce computational effort while maintaining stability.

4.7 Scheduled gain PID controller

This scheduled gain controller was tuned to reduce computational costs while maintaining PID controlled coupling error reduction standards of earlier experiments. The gains of the controller were changed manually in-between time-steps. It is called the "scheduled gain PID controller" because the gains are a function of time. Figure 4.16 depicts coupling error in the scheduled gain controller. This controller takes fewer time-steps than both the stable and unstable controller discussed earlier, however, it is stable. This suggests that this controller selected time-steps that were, to some degree, parallel to periods of pressure transience in the completed grid block of the reservoir simulation. Therefore, despite that the scheduled gain controller executed less time-steps than the unstable and stable controller (for $K_P = 1.0$ and $K_P = 0.1$, respectively), the time-steps it did calculate satisfactorily captured periods of transience.

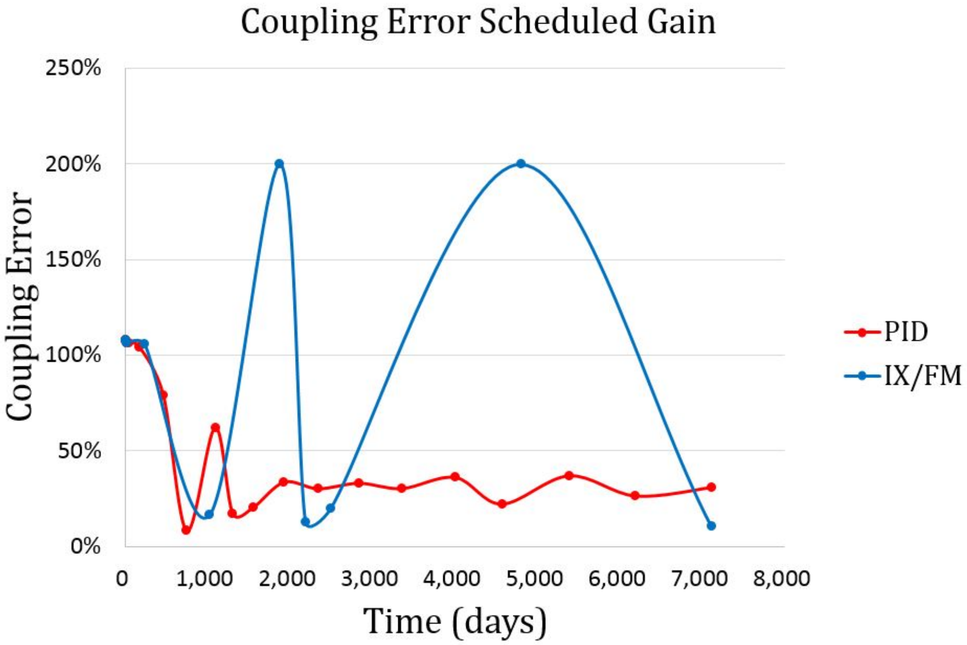

Figure 4.16: Coupling error in scheduled gain controller

Figure 4.17 below illustrates the gain schedule.

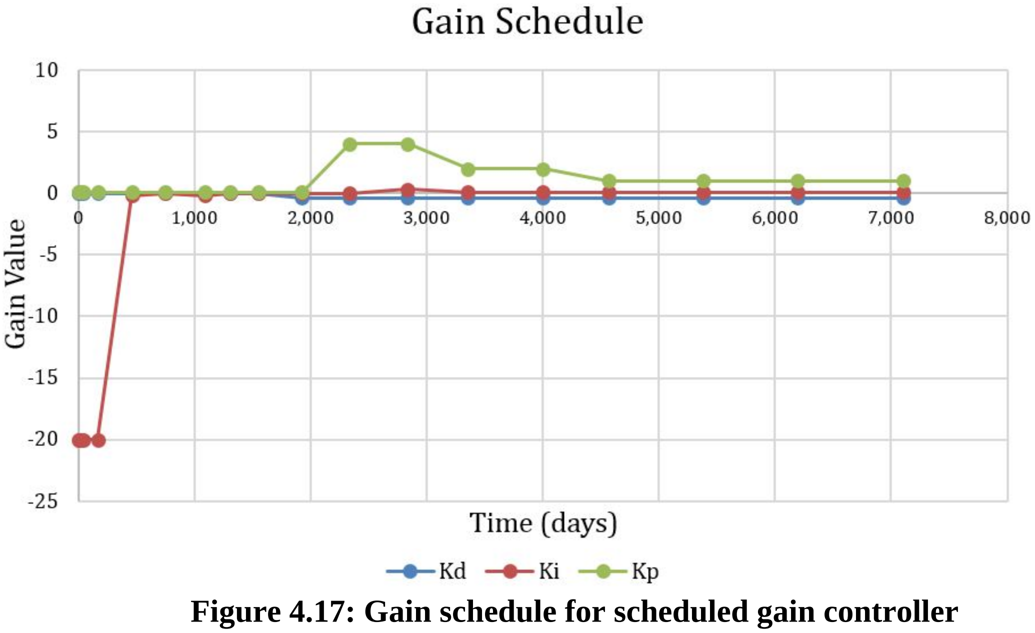

Figure 4.17: Gain schedule for scheduled gain controller

In addition to reducing error, the scheduled gain controller performed the simulation in 16.9 seconds, as compared to the 14.5 seconds the unstable base case took. Therefore, the PID controller was only about 17% slower than the unstable base case simulation. The scheduled gain PID controller serves as motivation for robust optimization of gain values, as well as the controller tolerance. To further illustrate the oscillation reduction capabilities, the tubing head pressure for the scheduled gain controller is illustrated below in figure 4.18.

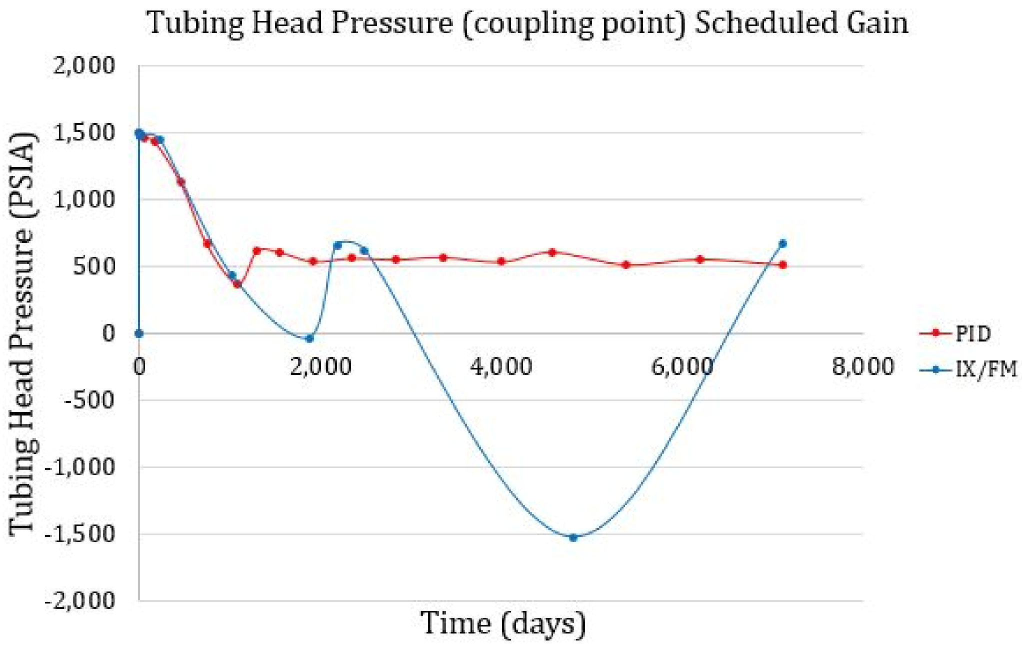

Figure 4.18: Tubing head pressure in scheduled gain controller

Allow the tubing head pressure is not oscillation free, the oscillations are much reduced as compared to the base case. In the following section, the scheduled gain controller and $K_p = 0.1$ controller will be compared to Schlumberger’s default coupling settings.

4.8 PID control in comparison to Schlumberger’s default coupling settings

This section compares two PID controlled experiments ($K_p = 0.1$ and scheduled gain controllers) to Schlumberger’s default coupling settings. The simulations in the following analysis do not have the same *TimeStepSizingControls* and *Convergence* criteria settings. As mentioned before, the PID controlled experiments feature relaxed settings that are described in chapter two. The simulation used as a standard or base case now is Schlumberger’s default settings in the scenario 1 simulation (PROD 1 only). In the following experiment, the efficacy of PID control will be compared to Schlumberger’s intended, and arguably optimal, coupling settings. In this sense, the following results illustrate the effectiveness of our PID control in comparison to standard Schlumberger commercial coupling performance. The following section compares the computational effort of the simulations (time required to complete the simulation, or CPU time) as well as the relative accuracy of the simulations (comparison of the cumulative productions).

Figure 4.19 compares the coupling error of Schlumberger’s standard time-stepping control and the PID controller with $K_p = 0.1$. Both simulations were normalized to 8,000 days. Figure 4.19 shows that Schlumberger's time-stepping algorithm is more effective at removing error.

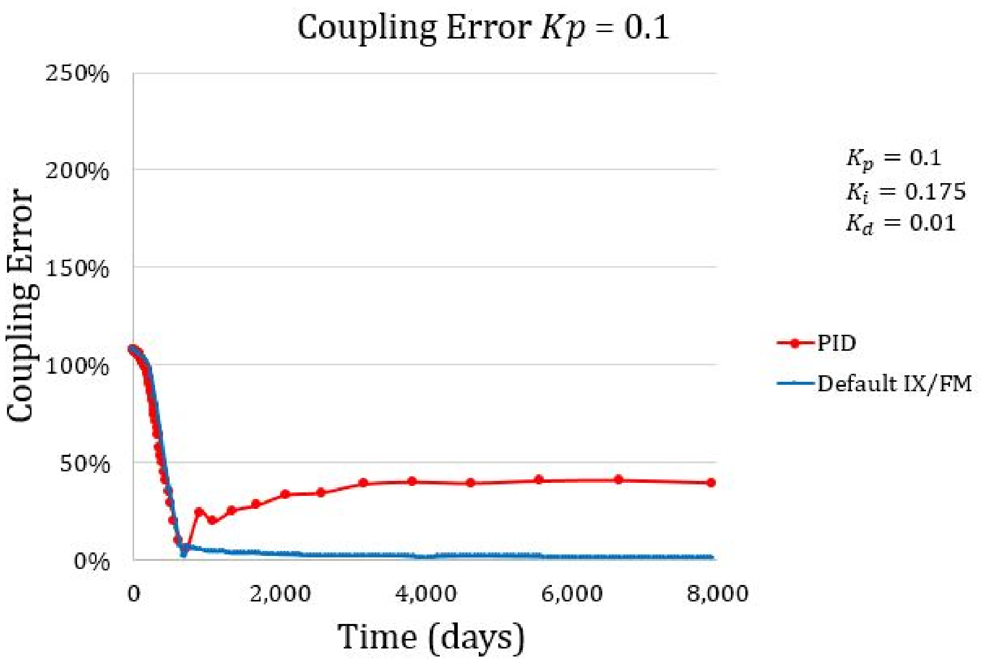

Figure 4.19: Coupling error for PID control $K_p = 0.1$ and default experiment

Schlumberger's default coupling settings executed 162 time-steps, as compared to the PID controller's 62 time-steps. The PID controller reduced error to a stable 45%, while Schlumberger's settings reduced error to a stable 2%. Figure 4.20 illustrates the commensurate tubing head pressure. We see that the PID predicts smaller values for tubing head pressures. This is because the surface system over estimates the VFP, and the reservoir does not deliver as well as it otherwise could in a system with less error. Tubing head pressure, illustrated in Figure 4.21, is also larger as calculated by Schlumberger's default settings. The tubing head pressure increases slightly from about 1,000 days to 2,500 days. This phenomenon is not observed in other pressure profiles in the default case.

This could be a result of the tubing head experiencing the constant boundary of the gathering line.

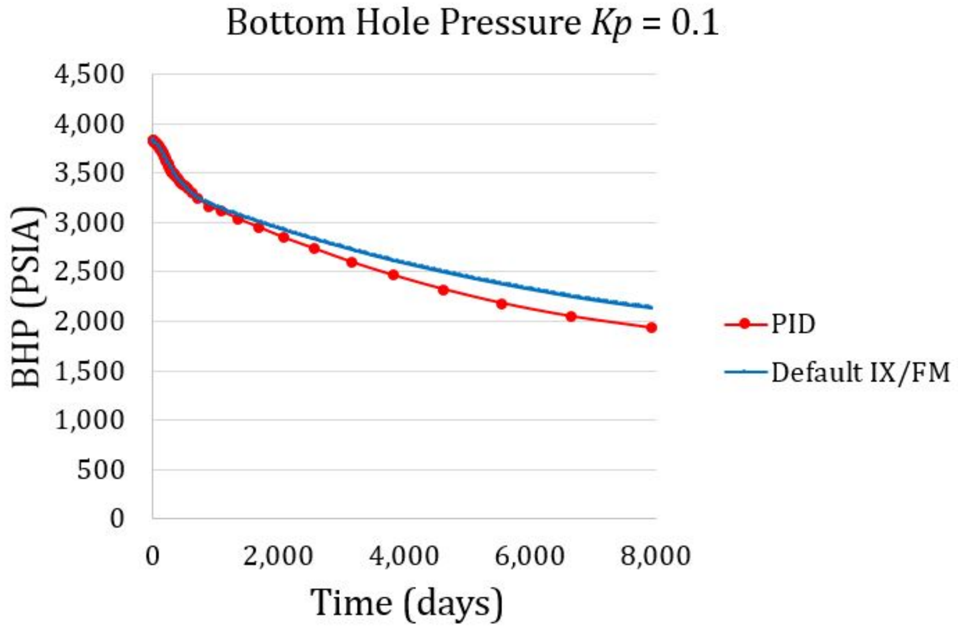

Figure 4.20: BHP for PID control $K_p = 0.1$ and default experiments

Other than the departures in tubing head pressure, it is interesting to note the somewhat linear departure between the PID experiment and the default experiment for bottom-hole pressure (significant) and cumulative production (insignificant). It is also interesting to note when the departures occurred: about 1,000 days. It is important to note that neither the production and bottom-hole pressure profiles behave as the tubing head pressure does.

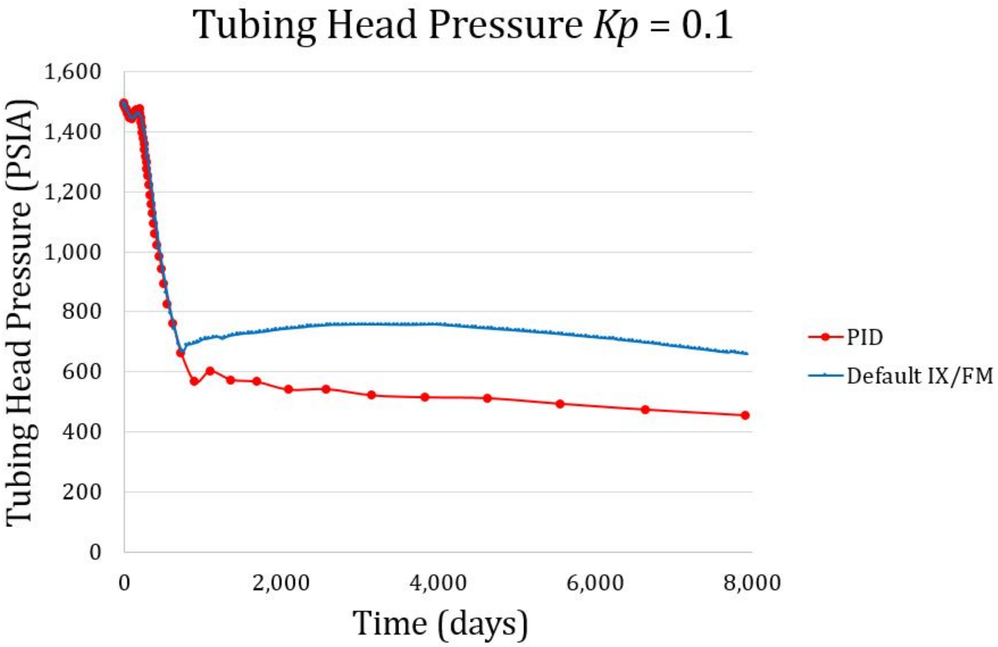

Figure 4.21: TBHP for PID control $K_p = 0.1$ and default experiments

Despite the disparities between the $K_p = 0.1$ controller and Schlumberger's default coupling simulation settings in error and pressure profiles, the PID controlled simulation only incurs 0.8% error in cumulative production. Figure 4.22 below illustrates this. If the object of the simulation is to determine project economics, such a small error in cumulative production may be acceptable. The most pertinent data is outlined in Table 4.5: this table illustrates that the PID controlled experiment is 45% faster.

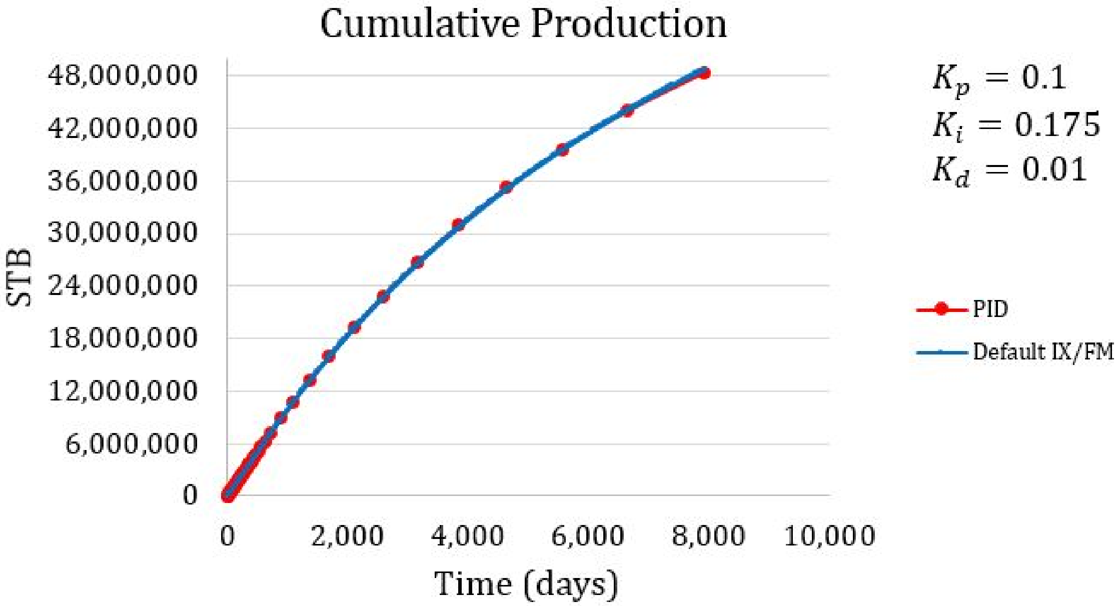

Figure 4.22: Cumulative production for PID $K_p = 0.1$ and default experiments

Optimization problems may require more robust time-stepping methods than PID control as presented. However, cumulative production values may be acceptable for investment benchmarking. The PID controller shown in figure 4.22 would thus considerably reduce the computational costs of the simulation, while maintaining fidelity in cumulative production.

	PID (K_p = 0.1)	Default Settings
Time Steps	62	162
Cumulative Production (STB)	48,402,300	48,794,500
Computational Effort (s)	37.49	54.55

Table 4.5: PID control and default settings performance parameters

If we consider the 0.8% error in cumulative production permissible, then the PID controlled experiment (K_p =0.1) is superior to Schlumberger's coupled time-stepping algorithm. Although the $K_p = 0.1$ controller may be considered considerably superior to the default simulation, the scheduled gain controller is far superior. Figure 4.23 compares error of the scheduled gain controller, as introduced in section 4.7, to the default coupled case.

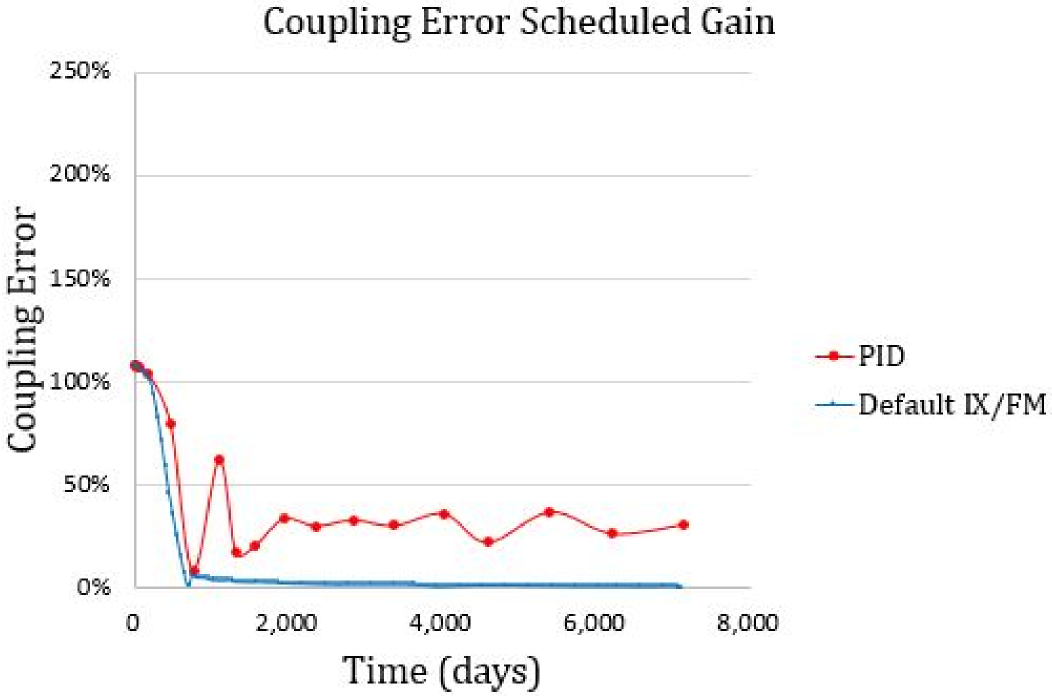

Figure 4.23: Coupling error for scheduled gain PID and default experiments

The scheduled gain controller was normalized to 7,100 days. Again, we observe that the PID does not reduce error as well as Schlumberger's default. The PID controller was not derived to reduce error, it was derived to reduce oscillation. The scheduled gain controller is more oscillatory that the previously discussed controller. Figure 4.24 illustrates the tubing head pressure for the scheduled gain controller and default case. The tubing head pressure profile is analogous to the earlier tubing head pressure behavior both with the PID controller and the default settings.

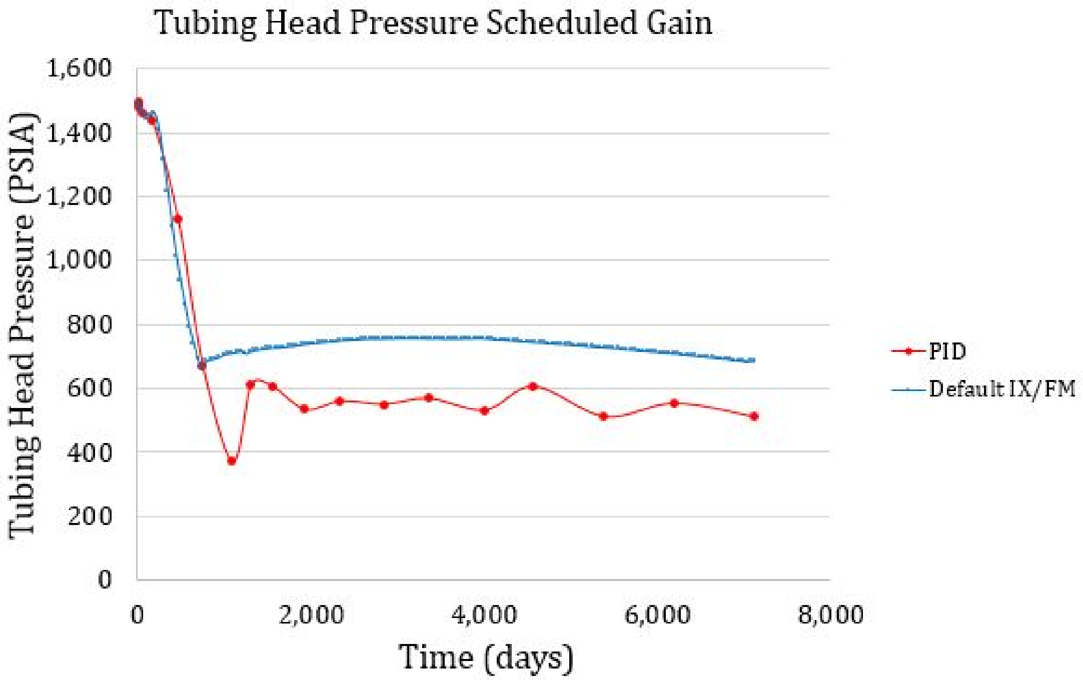

Figure 4.24: Tubing head pressure for scheduled gain PID and default experiments

Despite the oscillatory tubing head pressure curve calculated with PID control, this scheduled gain controller made a better match with the cumulative production value calculated by the default case. Figure 4.25 highlights the cumulative production curves, and table 4.4 highlights key performance parameters.

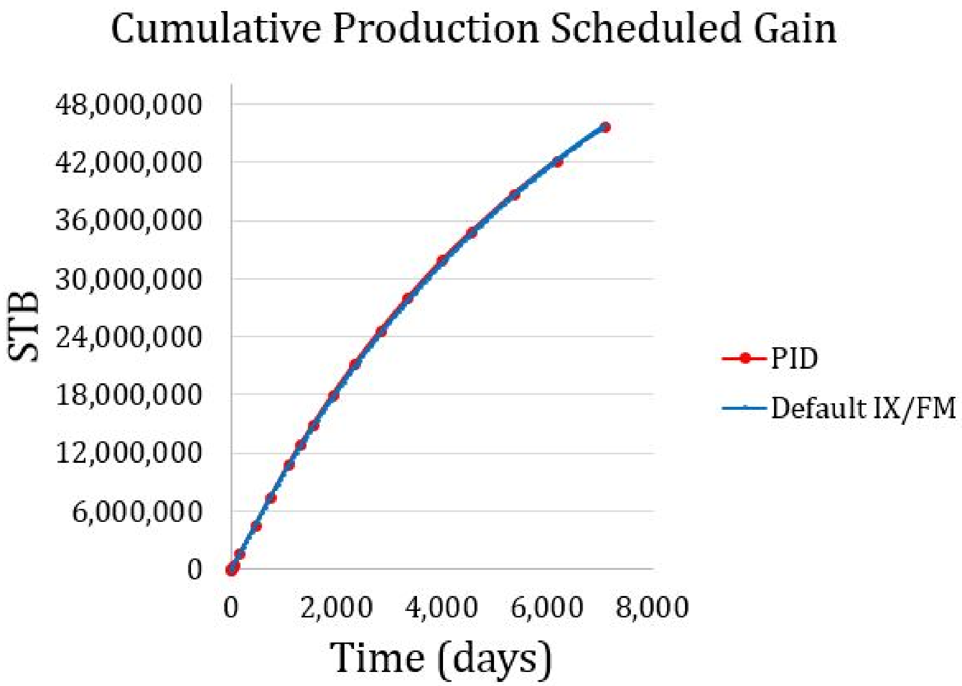

Figure 4.25: Cumulative production for scheduled gain PID and default

	PID (K_p = 0.1)	Default Settings
Time Steps	19	146
Cumulative Production (STB)	45,654,200	45,855,500
Computational Effort (s)	16.9	50.2

Table 4.6: Scheduled gain PID control and default settings performance parameters

In summary, the scheduled gain controller was three times faster than Schlumberger's default coupling simulation. In addition, there was less than a 0.5% error between the scheduled gain controller calculated cumulative production and the default simulation calculated cumulative production profile. Therefore, it can be argued that the scheduled controller is supremely superior to Schlumberger’s default settings when considering computational costs (simulation time) and cumulative production profile analysis. These results further motivate robust controller gain optimization through empirical methods in addition to well established methods like the Ziegler Nichols method. In addition, the tolerance in the PID controller should be optimized. Tolerance is treated as 1 in our experiments, and scales the integral term. Changing the tolerance should impact the value of error that the controlled simulation converges to. Improved error reduction may result in less oscillatory pressure and rate profiles.

4.9 Coupling error considerations

We noted earlier the definition of coupling error (equation 3.20). It is integral in our experiment. Coupling error is the input into the PID controller, and thus heavily influences the new, calculated time-steps in PID control. Coupling error was accessed during simulation using Schlumberger’s output query for the parameter, *NCEP*. The *RES1.PRT* was also analyzed to yield further insight into coupling error. It was discovered that the *Percent Mismatch* outputs were equivalent to *NCEP* and also an alternative moniker for coupling error. However, it was apparent in the *PRT* file that *Percent Mismatch*, or coupling error, was only roughly equal to the true value of coupling error as defined by Schlumberger, and as illustrated in equation 3.20. Individual reservoir (IX) and PIPESIM solutions at each time-steps were manually compared to confirm observations.

The “true coupling error” was determined for the scheduled gain controller experiment, and utilized the values *NetworkWellBoundary* and *Well*. *NetworkWellBoundary* and *Well* were the tubing head pressures (coupling point) as calculated by PIPESIM and INTERSECT, respectively. “True coupling error” was defined as equation 3.20. True coupling error and *NCEP* coupling error as output by INTERSECT (NCEP) for the scheduled gain controller, is outlined below in Figure 4.26 below.

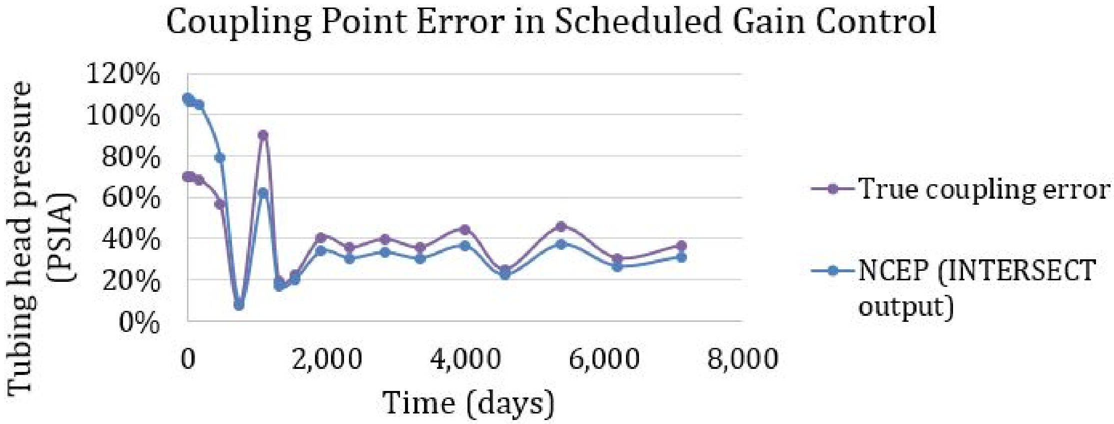

Figure 4.26: True coupling error and NCEP in scheduled gain control experiment

We see that true coupling error and *NCEP* do not vary with each other linearly. This is concerning, as there is no description in Schlumberger’s manuals on how coupling error as reported (*NCEP*) is actually calculated. In addition, all tests in this body of work utilized *NCEP* as an error input in the PID controller. Therefore, we did not determine if *NCEP* or true coupling error is the optimal controller input. In theory, true coupling error is the correct input into the controller. However, we did not test PID control using true coupling error. In the future, true coupling error should be utilized in PID calculations.

4.10 Spatial error control

Scenario two was tested with PID control to attempt to understand how various error control algorithms might work. Various error control algorithms were enabled because two coupling error outputs were calculated during simulation (for PROD1 and PROD2). The first coupling algorithm controlled the time-stepping with coupling error from PROD1 only. The second coupling algorithm controlled the time-stepping with coupling error from PROD2 only. The third and final coupling algorithm controlled the time-stepping with the average of PROD1 and PROD2 coupling error. We attempted to draw conclusions regarding error control and configuration of the injectors and producers. However, there were no clear results. [25] gains were used in this experiment.

Figures 4.27 and 4.28 illustrate error profiles for PID controlled simulation using PROD1 NCEP as the sole error input.

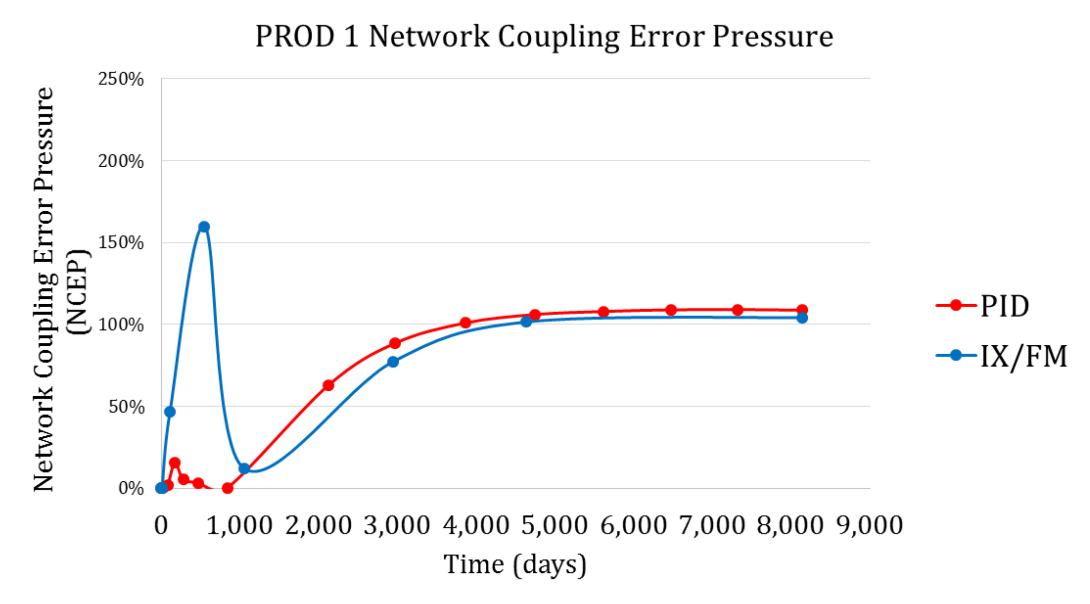

Figure 4.27: PROD1 NCEP in PID PROD1 error controlled simulation

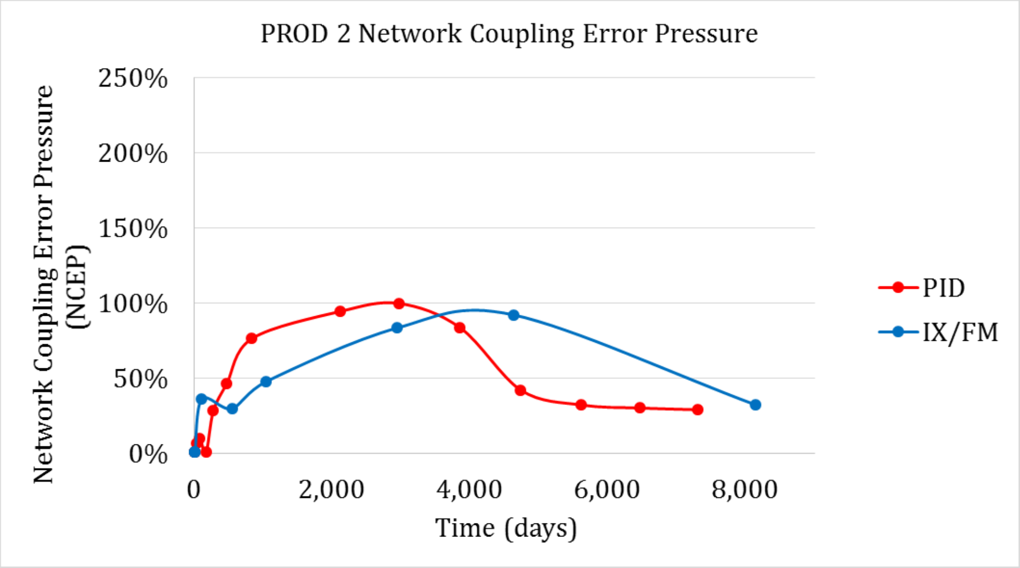

Figure 4.28: PROD2 NCEP in PID PROD1 error controlled simulation

Figures 4.29 and 4.30 illustrate error profiles for PID controlled simulation using PROD2 NCEP as the sole error input.

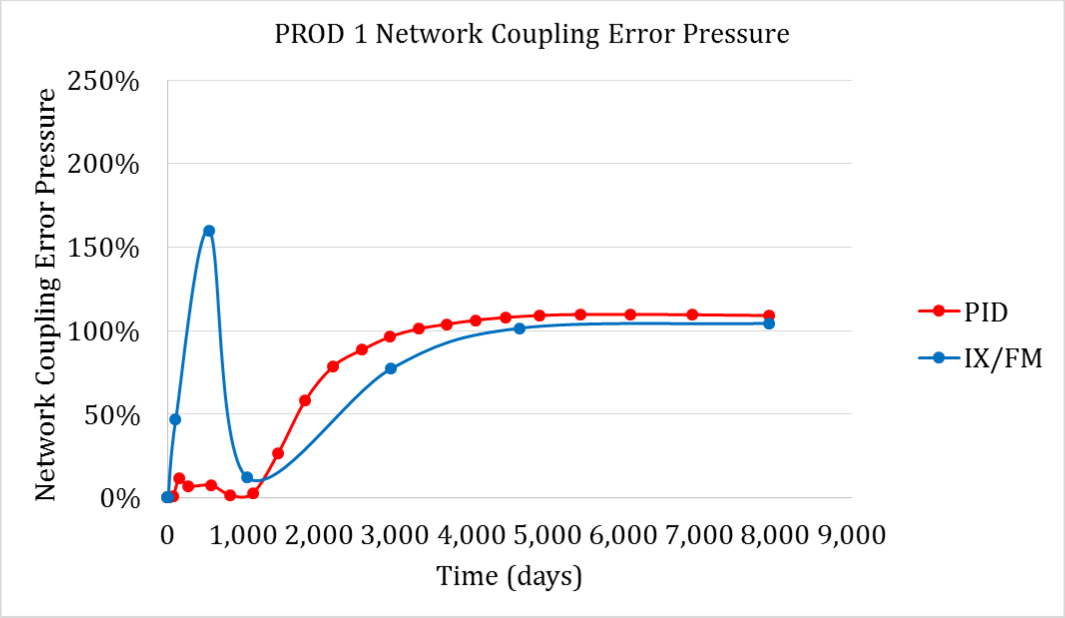

Figure 4.29: PROD1 NCEP in PID PROD2 error controlled simulation

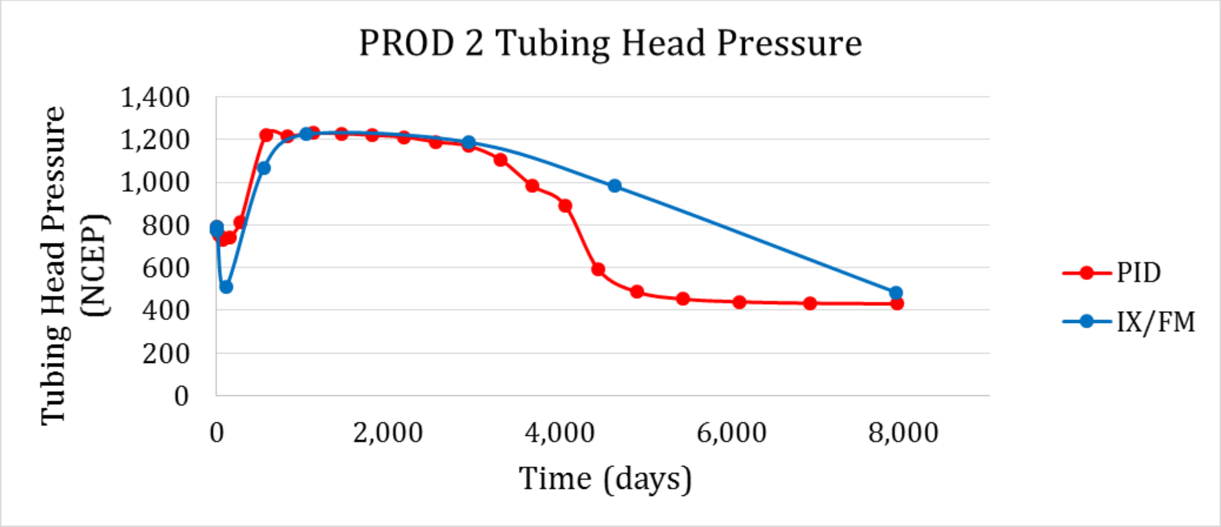

Figure 4.30: PROD2 NCEP in PID PROD2 error controlled simulation

Although there are variations in error for PROD1 and PROD2 error control schemes, there are no control cases for which an experimental spatial standard is clearly defined. That is, it is impossible to determine the effect the spatial configuration has on error in the current configuration without some base reference. The third coupling algorithm is not included in this discussion, as results likewise do not show any interest. In summary, a more robust experiment must be conducted if the influence of spatial well configuration on error can be further elucidated. Simplified configuration simulations may improve understanding.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

In conclusion, we showed that PID controlled adaptive time-stepping can effectively reduce oscillations in coupling error, pressure profiles and rate profiles in *Iteratively Lagged* (partially implicit) surface-subsurface coupled simulations. In addition, we showed that PID controlled simulation is far superior to Schlumberger's commercial coupling with regards to simulation time. Our best controller reduced computational costs by 300% with less than 0.5% error in cumulative production. Our time-stepping scheme thus immediately adds value to the engineer who is interested in production profiles of coupled systems. Various gains were tested, and the unstable controller highlighted that the size of the selected time-step is more important than the quantity of time-steps taken (with regards to error reduction and error stability). The scheduled gain controller that proved to be three times faster than Schlumberger's default commercial settings serves as motivation for additional, robust gain optimization such as Ziegler-Nichols. If possible, a scheduled gain algorithm should be developed. This scheduled gain algorithm would automatically select new gain values at the beginning of each time-step, as a function of past performance. Also, the type of controller used should be optimized. For example, the experimental PID controller derivative term approaches zero as subsequent error completely stabilizes. This behavior should be scrutinized with respect to error, pressure and rate profiles. PI control should be tested as well. A PI controller test would elucidate the effects of negating the derivative term. A controller might also be derived for the specific order of error and oscillations in our simulation. In addition, true coupling error should be substituted for NCEP in any error controlled time-stepping simulation. More rigorous analysis of PID controlled simulation should include detailed analysis of mass balance convergences across time (i.e. compare rates of reservoir and surface simulation at each time-step). Our simulations did not violate Schlumberger's internal mass balance convergence criteria; however, a more detailed review of mass balance convergence performance (between the reservoir and the surface network) would further elucidate the mechanism of PID controlled coupling. A more detailed mass balance convergence analysis may also elucidate if PID calculated time-steps are optimized to capture transient pressure regimes. If time-step action is not associated with severe changes in pressure and rates, opportunity for improvement would be highlighted. In such a scenario, the controller might be tuned to actuate when pressure and rate changes drastically.

In addition, a standard PID, PI and P controller should be tested. Also, we specified a tolerance of one in our PID controller. This value of one was suggested in the literature for error control purposes. Optimizing the tolerance could further improve controller performance.

We initially proposed to use PID control to reduce Newton-Raphson iterations during simulation. In theory, the controller could use an initial Newton-Raphson solution to quickly predict a subsequent Newton-Raphson iteration solution in the case that the first iteration does not converge to a user specified tolerance in simulation. Figure 4.31 illustrates our initial idea. We attempted to implement this idea in INTERSECT, however it was impossible. The current configuration of INTERSECT does not allow the user to access any Newton-Raphson solutions, let alone actively manipulate the Newton-Raphson iterations. We believed that this approach could be used to reduce the number of Newton-Raphson iterations needed for a solution to converge within a tolerance. This would reduce the computational cost of each time-step. [1] discussed the basis of this idea, and we attempted to fully develop and implement this idea.

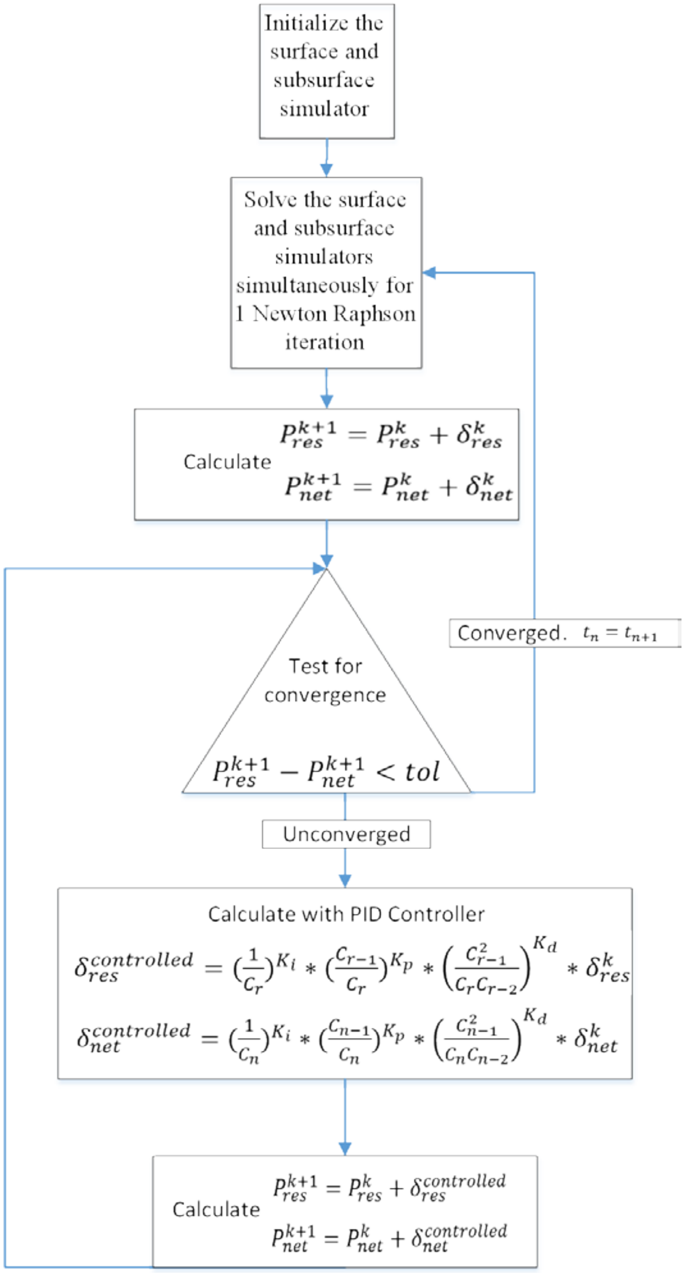
Initialize the surface and subsurface simulator
Solve the surface and subsurface simulators simultaneously for 1 Newton Raphson iteration
Calculate $P_{res}^{k+1} = P_{res}^{k} + \delta_{res}^{k}$ $P_{net}^{k+1} = P_{net}^{k} + \delta_{net}^{k}$
Converged. $t_n = t_{n+1}$
Test for convergence
$P_{res}^{k+1} - P_{net}^{k+1} < tol$
Unconverged
Calculate with PID Controller
$\delta_{res}^{controlled} = (\frac{1}{C_r})^{K_i} * (\frac{C_{r-1}}{C_r})^{K_p} * \left(\frac{C_{r-1}^2}{C_r C_{r-2}}\right)^{K_d} * \delta_{res}^{k}$
$\delta_{net}^{controlled} = (\frac{1}{C_n})^{K_i} * (\frac{C_{n-1}}{C_n})^{K_p} * \left(\frac{C_{n-1}^2}{C_n C_{n-2}}\right)^{K_d} * \delta_{net}^{k}$
Calculate $P_{res}^{k+1} = P_{res}^{k} + \delta_{res}^{controlled}$ $P_{net}^{k+1} = P_{net}^{k} + \delta_{net}^{controlled}$

Figure 5.1: PID control in Newton-Raphson iterations

REFERENCES

1. Akakpo, Dany, and Eduardo Gildin. "A Control Perspective to Adaptive Time-Stepping in Reservoir Simulation". *SPE Reservoir Simulation Conference.* 20-22 February, Montgomery, Texas, USA. Society of Petroleum Engineers. 2017. 1-18. Print.
2. Beggs, D. H., and J. P. Brill. "A Study of Two-Phase Flow in Inclined Pipes." *Journal of Petroleum Technology* 25.05 (1973): 607-13. Print.
3. Byer, Thomas J., Michael G. Edwards, and Khalid Aziz. "Preconditioned Newton Methods for Fully Coupled Reservoir and Surface Facility Models". *SPE Annual Technical Conference and Exhibition.* 27-30 September, New Orleans, Louisiana. Society of Petroleum Engineers, 1998. Print.
4. Cao, H., et al. *A Fully Coupled Network Model, Practical Issues and Comprehensive Comparison with Other Integrated Models on Field Cases*. SPE Reservoir Simulation Symposium, 23-25 February, Houston, Texas, USA: Society of Petroleum Engineers, 2015. Print.
5. Coats, B. K., et al. *A Generalized Wellbore and Surface Facility Model, Fully Coupled to a Reservoir Simulator*. SPE Reservoir Simulation Symposium, 3-5 February, Houston, Texas: Society of Petroleum Engineers, 2003. Print.
6. Dempsey, J. R., et al. "An Efficient Model for Evaluating Gas Field Gathering System Design." *Journal of Petroleum Technology* 23.09 (1971): 1067-72. Print.

7. E. Hairer, and G. Wanner. *Solving Ordinary Differential Equations II*. Second revised edition with 137 figures. Ed. DE: Springer Verlag, 1996. Springer Series in Computational Mathematics Web.

8. Eidson, B. L., J. Y. Hung, and R. Mark Nelms. "An Experimental Evaluation of the PID Controller Represented by the Delta Operator". Orlando Fl. IEEE, 2012. 1-6. Print.

9. Emanuel, Alan S., and Jon C. Ranney. "Studies of Offshore Reservoir with an Interfaced Reservoir/Piping Network Simulator." *Journal of Petroleum Technology* 33.03 (1981): 399-405. Print.

10. Ertekin, Turgay, J. H. Abou-Kassem, and G. R. King. *Basic Applied Reservoir Simulation*. Richardson, US: Society of Petroleum Engineers, 2000. Web.

11. Fang, W. Y., and K. K. Lo. "A Generalized Well Management Scheme for Reservoir Simulation." *SPE Reservoir Engineering* 11.02 (1996): 116-9. Web.

12. Gao, Mengdi. "Reservoir and Surface Facilities Coupled through Partially and Fully Implicit Approaches." Texas A&M, 2014. Print.College Station, TX.

13. Ghorayeb, Kassem, et al. "A General Purpose Controller for Coupling Multiple Reservoir Simulations and Surface Facility Networks". *SPE Reservoir Simulation Symposium.* 3-5 February, Houston, Texas. Society of Petroleum Engineers, 2003. Print.

14. Gustafsson, K. "A PI Stepsize Control for the Numerical Solution of Ordinary Differential Equations." *BIT Numerical Mathematics* 28.2 (1988): 270-87. Print.

15. Gustafsson, K. "A PI Stepsize Control for the Numerical Solution of Ordinary Differential Equations." *BIT Numerical Mathematics* 28.2 (1988): 270-87. Print.

16. Guyaguler, Baris, and Kassem Ghorayeb. "Integrated Optimization of Field Development, Planning, and Operation". *SPE Annual Technical Conference and Exhibition.* 24-27 September, Houston, TX. Society of Petroleum Engineers, 2006. Print.

17. Guyaguler, Baris, et al. "Near-Well-Subdomain Simulations for Accurate Inflow-Performance-Relationship Calculation to Improve Stability of Reservoir/Network Coupling." *SPE Reservoir Evaluation & Engineering* 14.05 (2011): 634-43. *OnePetro.* Web.

18. Hayder, Ehtesham M., Matthew Dahan, and Mubarak Nasser Dossary. *Production Optimization through Coupled Facility-Reservoir Simulation*. Houston, TX: Society of Petroleum Engineers, 2006. Print.

19. Hepguler, Gokhan, Santanu Barua, and Wade Bard. "Integration of a Field Surface and Production Network with a Reservoir Simulator." *SPE Computer Applications* 9.3 (1997): 88-92. Print.

20. Jensen, Ole. "An Automatic Time-step Selection Scheme for Reservoir Simulation". Dallas, Texas. Society of Petroleum Engineers, Sep 21, 1980. Print.

21. Katz, Donald L. *Handbook of Natural Gas Engineering*. New York: McGraw-Hill, 1959. Print.

22. Liang, Jialing, and Barry Rubin. "A Semi-Implicit Approach for Integrated Reservoir and Surface-Network Simulation". *SPE Reservoir Simulation Symposium.* The

Woodlands, Texas, USA. 18-20 February: Society of Petroleum Engineers, 2014. Print.

23. Litvak, M. L., and B. L. Darlow. "Surface Network and Well Tubinghead Pressure Constraints in Compositional Simulation".12-15 February, San Antonio, Texas. Society of Petroleum Engineers, 1995. 325-355. Print.
24. Mathworks. *Matlab*. R2014b Vol. Natick, MA: 2014. Print.
25. Moller, Matthias. "Time Stepping Methods." Athens Course. TU Delft, Delft. 19 Nov. 2014. Lecture.
26. O'Dwyer, Aidan. *Handbook of Pi and Pid Controller Tuning Rules (3rd Edition)*. 3rd ed. ed. GB: Imperial College Press, 2009. Web.
27. Ogazi, A. I., et al. "Severe Slugging Control through Open Loop Unstable PID Tuning to Increase Oil Production". Cannes, France. BHR Group, 2009. 17-29. Print.
28. Peaceman, D. W. "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation." *Society of Petroleum Engineers Journal* 18.03 (1978): 183-94. Web.
29. Russell, T. F. "Stability Analysis and Switching Criteria for Adaptive Implicit Methods Based on the CFL Condition". *SPE Symposium on Reservoir Simulation.* 6-8 February, Houston, Texas. Society of Petroleum Engineers. 1989. 97-107. Print.
30. Schiozer, D. J., and Khalid Aziz. "Use of Domain Decomposition for Simultaneous Simulation of Reservoir and Surface Facilities". *SPE Western Regional Meeting.* 23-25 March, Long Beach, California. Society of Petroleum Engineers, 1994. Print.
31. Schlumberger. *The Network Option*. 1987. Print.

32. Schlumberger Information Systems. *INTERSECT Technical Description 2016.1*., Print.

33. Schlumberger Information Systems. *INTERSECT User Guide 2016.1*., Print.

34. Schlumberger Information Systems. *PIPESIM User Guide 2015.1*. 2015. Print.

35. Soderlind, Gustaf, and Lina Wang. "Adaptive Time-Stepping and Computational Stability." *Journal of Computational and Applied Mathematics* 2.3 (2006): 225-43. Web.

36. Startzman, R. A., et al. "Computer Combines Offshore Facilities and Reservoir Forecasts." *Petroleum Engineer* (1977): 65-74. Web.

37. Trick, M. D. "A Different Approach to Coupling a Reservoir Simulator with a Surface Facilities Model". Houston, TX. Society of Petroleum Engineers, Mar 15, 1998. Print.

38. Valli, A. M. P., G. F. Carey, and Coutinho, A. L. G. A. "Control Strategies for Time-step Selection in Simulation of Coupled Viscous Flow and Heat Transfer." *Communications in Numerical Methods in Engineering* 18.2 (2002): 131-9. *CrossRef*. Web.

39. Zhang, Yanbin, Gaurav Seth, and Jianping Chen. "A Novel IPR Calculation Technique to Reduce Oscillations in Time-Lagged Network-Reservoir Coupled Modeling using Analytical Scaling and Fast Marching Method". *SPE Reservoir Simulation Conference*. 20-22 February, Montgomery, Texas, USA. Houston, TX: Society of Petroleum Engineers, 2017. Print.

40. Zhang, Yanbin, et al. *Fast-Marching Methods for Complex Grids and Anisotropic Permeabilities: Application to Unconventional Reservoirs*. Houston, TX: Society of Petroleum Engineers, 2013. Print.

41. Zhu, Ding, et al. *Petroleum Production Systems*. Upper Saddle River, NJ: Pearson Prentice Hall, 2012. Print.