

Integrated Field Modeling

Bamshad Nazarian

A Dissertation for Partial Fulfillment of Requirements for the Degree of
Doktor Ingeniør

Department of Petroleum Engineering and Applied Geophysics

Norwegian University of Science and Technology

December 2002

to Anahita

Acknowledgment

I want to express my sincere gratitude and appreciations to my supervisor professor Michael Golan for his extraordinary arrangements and countless helps that made this study possible. I would also like to thank him for numerous valuable discussions we had while working on this project.

I would also like to give my special thanks to professor Curtis H. Whitson for all I learnt from him during my stay at NTNU. His comments about writing style and his help with EOS calculations and the reservoir model used in optimization case studies are gratefully acknowledged.

The financial support from TotalFinaElf was instrumental in finishing this work. This support is sincerely acknowledged. I am indebted to professor Jon Kleppe for his kind assistance in arranging the financial support. His help is highly acknowledged.

Helge Hole and Andy Howell both from Hyprotech provided valuable support and materials regarding HYSYS extensibility. Their assistance is greatly acknowledged.

I would like to thank dear friends and colleagues Mohammad Faizul Hoda, Kameshwar Singh, Hu Bin, and Vidar Haugse for countless interesting and valuable discussions I had with them in the past five years.

I thank my wife Gunn Hilde for her encouragements, understanding, and support during the preparation of this thesis. Finally, I wish to thank my parents for all they provided me and for their continuous support.

Summary

This research project studies the feasibility of developing and applying an integrated field simulator to simulate the production performance of an entire oil or gas field. It integrates the performance of the reservoir, the wells, the chokes, the gathering system, the surface processing facilities and, whenever applicable, gas and water injection systems.

The approach adopted for developing the integrated simulator is to couple existing commercial reservoir and process simulators using available linking technologies. The simulators are dynamically linked and customized into a single hybrid application that benefits from the concept of open software architecture. The integrated field simulator is linked to an optimization routine developed based on the genetic algorithm search strategies. This enables optimization of the system at field level, from the reservoir to the process. Modeling the wells and the gathering network is achieved by customizing the process simulator.

This study demonstrates that the integrated simulation improves current capabilities to simulate the performance of an entire field and optimize its design. This is achieved by evaluating design options including spread and layout of the wells and gathering system, processing alternatives, reservoir development schemes, and production strategies.

Effectiveness of the integrated simulator is demonstrated and tested through several field-level case studies that discuss and investigate technical problems relevant to offshore field development. The case studies cover topics such as process optimization, optimum tie-in of satellite wells into existing process facilities, optimal well location, and field layout assessment of a high pressure high temperature deepwater oil field.

Case study results confirm the viability of the total field simulator by demonstrating that the field performance simulation and optimal design were obtained in an automated process with reasonable computation time. No significant simplifying assumptions were required to solve the system and tedious manual data transfer between simulators, as conventionally practiced, was avoided.

Chapter 1 discusses incentives for creating a software platform for total field simulation. It justifies the adopted integration approach and discusses the reasons behind selecting the chosen simulators.

Chapter 2 describes the methods used in development of the hybrid application and discusses its capabilities and limitations. This chapter explains important issues to be considered in using linked applications such as the need and measures to harmonize fluid characterization between process and reservoir simulators. This chapter also explains how the process simulator has been customized to simulate wells and gathering network.

Chapter 3 addresses the problem of field optimization and explains the development of a genetic algorithm search routine for global optimization. It demonstrates and discusses performance of the search program through several optimization case studies.

Chapter 4 is about field development planning. Using a case study, entire program is tested by studying the development of a deepwater prospect. A difficult reservoir and demanding marine environment subject the design to a range of constraints and limitations. Five development scenarios have been identified. The program is used to assess and rank the scenarios according to conventional assessment parameters such as cumulative production, costs, complexity, operability, and Net Present Value (NPV).

Chapter 5 summarizes the results, observations and conclusions and provides suggestions for future work.

Appendix A includes information regarding Automation techniques used in development of the hybrid application. Several code examples are used to clarify the method of coupling various applications through Automation.

Appendix B describes working procedure of the executive program. The main dialog boxes with which the user interacts have been depicted and discussed in this appendix. **Appendix C** is devoted to some detailed discussions regarding flow regime determination in compositional systems. The flow regime routine that has been used in this study is explained in this appendix. **Appendix D** presents selected results to allow independent control of the simulation method used in solving the network example of **Chapter 2**.

Contents

Acknowledgment	i
Summary	iii
1 Introduction	1
1.1 Background and objectives	1
1.2 A platform	6
1.3 Integrated system requirements	7
2 The field simulator	9
2.1 Introduction	9
2.2 Background	11
2.3 Available integration techniques	13
2.4 The developed tool	15
2.5 Well model	20
2.5.1 Beggs and Brill pressure gradient	22
2.5.2 Gregory Aziz Mandhane pressure gradient	22
2.5.3 Two-Fluid model (OLGAS) pressure gradient	23
2.6 Gathering network model	24
2.6.1 Introducing the HYSYS.process as a network solver	25
2.6.2 Mixer operation	26
2.6.3 Adjust logical operation	27
2.6.4 Solution of converging pipe networks	27

2.7	Reconciliation of Eclipse and HYSYS simulations	31
2.7.1	EOS tuning in Eclipse and HYSYS	32
2.7.2	The Cubic Equation of State	33
2.7.3	The Peng-Robinson Equation of State	36
2.7.4	The proposed tuning method	37
2.7.5	Remarks	43
3	Optimization	45
3.1	Introduction and background	45
3.2	A brief look at available optimization techniques	48
3.3	Genetic algorithms	51
3.3.1	Structure of simple genetic algorithms	52
3.3.2	Major advantages of genetic algorithms	55
3.3.3	Vocabulary	57
3.3.4	Special operators in genetic algorithms	58
3.3.4.1	Niche and speciation	58
3.3.4.2	Elitism	62
3.3.4.3	Shuffling and inversion operator	62
3.3.4.4	Micro operators	62
3.3.5	Numerical example	63
3.3.5.1	Optimization problem	63
3.3.5.2	Representation	64
3.3.5.3	Initial population	66
3.3.5.4	Selection	67
3.3.5.5	Crossover	69
3.3.5.6	Mutation	71
3.4	VisualGene	73
3.4.1	A test problem for VisualGene	74
3.4.1.1	The test function	74
3.4.1.2	The VisualGene solution	76

3.5	Case studies	78
3.5.1	Case study 1 – Optimization of separator pressures	79
3.5.2	Case study 2 – Tying satellite wells into existing facilities	84
3.5.3	Case study 3 – Optimal well location	89
3.5.4	Remarks	92
3.6	Constrained optimization	94
3.6.1	Handling constraints	95
3.6.2	Available techniques	95
3.6.3	Case study 4 – Optimal well location with water production constraint	98
3.6.4	Handling constraints and discontinuous surfaces in VisualGene	101
3.6.5	Remarks	104
4	Production system analysis	109
4.1	Introduction	109
4.2	Reservoir unit	110
4.3	Network	112
4.4	Riser, processing plant and transfer line	115
4.4.1	First scenario	116
4.4.2	Second scenario	117
4.4.3	Third scenario	118
4.4.4	Fourth scenario	119
4.4.5	Fifth scenario	120
4.5	Comparison of alternatives	121
4.6	Determination of flow regimes	126
4.7	Remarks	128
4.8	General observations regarding production from deepwater prospects	129

5 Conclusions	133
5.1 Observations	133
5.2 Overall conclusions	134
5.3 Recommendations for future work	135
Nomenclature	137
References	139
Appendix A: Automation	151
Appendix B: Working with LinkControl	161
Appendix C: Flow regime determination	167
Appendix D: HYSYS simulation results	183

Chapter 1

Introduction

1.1 Background and objectives

Conceptual field development studies require assessment of large number of design parameters that constitute a field development plan. This is true for any field development case, yet, it is particularly important in standalone offshore fields where major development decisions need to be taken upfront with limited flexibility for changes later on.

Major decision parameters in development plans are:

- Number of production and injection wells.
- Well placement.
- Connectivity arrangement between wells and pay-zones (well-reservoirs interface).
- Drilling schedule.
- Reservoir off-take schedule.
- Wellhead spread (cluster versus satellite arrangement).
- Wells trajectory.
- Tubing size.
- Artificial Lift.
- Field and gathering system layout.
- Flowline size.
- Production riser configuration, size and riser-base gas lift.
- Seabed processing (separation, compression, pumping).
- Temperature control of the gathering system.
- Pressure and temperature of first stage separator.
- Stage separation configuration (number of stages, pressures and temperatures).
- Products handling capacity.
- Export streams composition.

- Export schemes (delivery points, and delivery specifications).
- Injection capacity and relevant compression/pumping arrangements.

Although certain design decisions are related to the architecture and the facility layout, most of the decisions can be expressed quantitatively and can be considered as design variables in a flow-process problem. Some of the variables – like tubing sizes – assume discrete values, other variables are continuous in nature – like reservoir pressure or producing gas oil ratio – but can be treated as discrete quantities in the design.

The production system is essentially a flow system fed by the reservoir at one end and exit of product or disposal streams at the other end. Though, there is quite a lot of dynamism and rapid transience in the production process, field design concentrates on the steady state or pseudo-steady state aspects of the production and views the transience merely as noise. Design variables are basically parameters in a complex multi-entry, multi-exit flow-process system whose performance can be calculated given values of the design parameters.

Identifying, screening, assessing and ranking field design options are strongly dependent on the changes in the reservoir, which take place during the life of the field. Depletion related variations imply relatively slow changes in pressure, water production rate, composition of produced fluids, and free gas production. These variations lead to changes in the stream entering the wells, which in turn causes variations in the gathering network and the processing system. Therefore, modeling reservoir behavior is a major part of the design. Truly, certain aspects of time dependent reservoir behavior can be treated by simple material balance calculations or decline curves. Yet, a 3-D reservoir simulator is essential to assess important design parameters such as well placement.

A study to determine the effect of various design parameters requires a global system assessment. At field level, many of the design variables are interrelated and design considerations traverse several engineering disciplines. For example, decision on well location is related to the time of water breakthrough in the well and start of loading the wells, the gathering system and the processing plant with water. Therefore, studying the performance of the system with an engineering tool that integrates all the existing subsystems appears to be necessary.

Such an engineering tool should allow evaluation of the entire system performance, with many combinations of design parameters. Usually a large number of possible scenarios with wide ranges of design parameters should be assessed. Experience, common sense, and intuition can reduce considerably the range of parameters and their combinations and thus the number of cases to be assessed. Yet, the computation effort is so large that a systematic approach, supported with automated calculation is needed in evaluation process.

An additional aspect of the conceptual design affects the number of evaluations needed. This is the uncertainty involved primarily in reservoir description of features such as: anisotropy, units and zonal communication, pressure support, and etc. Reservoir uncertainties at the conceptual design stage are very large. It is inherent to field planning that major conceptual decisions need to be taken upfront when the uncertainties regarding reservoir behavior are at maximum level. Uncertainties are gradually narrowed during production of the field due to monitoring and interpretation of reservoir response. The impact of uncertainties and the risks they present to the viability of a development plan need to be quantified and assessed as part of the decision-making process. This requires additional system evaluations where the uncertain variables assigned several values representing the range of uncertainty and the likelihood of occurrence.

Assessing or ranking the evaluated concepts and determining the optimum combination of design parameters require a well-define ranking or assessment criterion. Mathematically, the criterion is an objective function which is optimized by varying the value of the decision variables. Optimization function in a conceptual field design can be, for example, constrained maximum recovery over a certain production period, or maximum profit expressed as net present value. A given set of defined constraints, such as maximum allowable water production, or maximum gas injection rate, limits the range of the control variables and adds to the complexity of the optimum search. The search for optimum in a multivariate optimization case requires many evaluations of objective function, and thus multiple evaluation of the entire system performance. The complexity and the multivariate nature of field production design, suggest that concept optimization should be conducted automatically using an optimum search program that is linked automatically to the integrated field simulator.

In review, the integrated field conceptual design and concept optimization greatly benefit from the capability to simulate the entire production system with an integrated field simulator and integrated optimization strategy. Developing such engineering capabilities is the subject of this research project.

Presently, there is no universal working process in modeling, simulating and optimizing the entire field during the conceptual design phase. Components of the system are traditionally treated independently in both design and operation. In the conceptual design phase of the project, each component typically models its share of the system based on some local rather than global criteria and generates results for use in other subsystems. The results of such deterministic analysis are manually transferred between the subsystems in a time-consuming and error-prone manner.

This process is normally done either in a sequential or a parallel manner. In the sequential engineering process, outcome of an upstream engineering activity is transferred as an input to a subsequent activity down the design chain. As the simulators of different subsystems often use different data structure, time scales, standards, and etc., transfer of information between the simulators compromises the quality of information.

In the parallel engineering process, each subsystem uses a simplified or an approximate model of the other relevant subsystems to generate the needed boundary conditions and to create a local model of the entire system. For instance, it is common to use a simplified reservoir model (tank model) in gathering system simulators. Simplified well, gathering and process models are built in the reservoir simulators and simplified reservoir functions are used in process and topside facility simulators. This is a source for many inconsistencies at field level.

Efforts to develop and use integrated field simulators in the last twenty years have been conducted in the following directions:

- Developing a comprehensive simulator from basic principles using shared computation routine for simulating the entire system. These developments have amounted to a gigantic effort that is still far from reaching completeness and universality.

- Customizing existing reservoir simulators to include routines to simulate the wells, gathering system and the process. This approach has yielded an option to improve the simulation by including facilities to simulate the production of various reservoir units, to account for wells and surface constraints, and in certain cases to include simple separation schemes. This approach is very good for field studies where reservoir performance is the main concern. In conceptual field designs, particularly offshore and sub-sea fields, there are numerous design decisions that are difficult to address in an “extended reservoir simulator”.

Successes of recently developed simulation methods in related disciplines, suggest adaptation of similar strategies to devise new methods to develop integrated field simulators that handle petroleum engineering problems more effectively. The idea of integration and open software architecture through development of plug-and-play software components is pursued by the process industry through European Community sponsored CAPE-OPEN and Global CAPE-OPEN projects (Braunschweig, 1999). The central idea behind these projects inspired the development work in this research to apply the concept of integrated simulation to petroleum engineering applications.

The goals of this study are:

- 1) To demonstrate that a stable integrated simulation system can be generated using presently available software technology.
- 2) To prove that such a system can effectively simulate and solve field scale problems where current simulation and optimization practices need improvement.

The idea of integrated simulation in petroleum engineering is not new and there have been numerous attempts in coupling process and reservoir simulators with various levels of sophistication (Pieters 1995, Breaux 1984, Trick 1998, Johansen 2000, and Lamey 1999). However, there are two novel features in the proposed integrated field simulator not reported before. These are:

1. **The linking method between the reservoir unit and the rest of the system.** The proposed linking method has resulted in a stable communication with the chosen 3-D reservoir simulator without restricting any of its functionalities.
2. **The open software architecture.** Open software architecture provides considerable flexibility in use and modification of the integrated system. This characteristic is brought to the system partly by the selected process simulator and partly by the Visual Basic programming language with which the system executive program and the communication protocols have been developed.

The open software structure made it possible to integrate a sophisticated search routine that is used to carry out optimization studies at various parts of the system. Combination of capabilities of the integrated search routine and seamless communication with the 3-D reservoir simulator enabled the system to carry out interesting optimization studies at reservoir level.

1.2 A platform

The product of the integration method introduced in this study is presented in the form of a software platform, based on which new integrated simulators can be developed. As an example of such a simulator, a prototype is created by connecting a reservoir simulator and a process simulator.

Among the available reservoir simulators, Eclipse simulator developed by GeoQuest had the characteristics required in a simulator that can be used in an integrated system because of these reasons:

1. At present, Eclipse is widely used in oil industry and is considered as the industry's standard reservoir simulator.
2. By developing Open-Eclipse protocols, GeoQuest changed Eclipse from a batch program controlled solely by the content of its input file to an interactive program that could be linked to other applications programmatically (Eclipse documentation 2002, and Open-Eclipse documentation 2002).

HYSYS simulator developed by Hyprotech is the software of choice to simulate the surface facilities (HYSYS documentation, 2000). HYSYS is considered to be one of the leading software packages in the field of integrated simulation environment available to process industry. HYSYS supports several integration techniques and is completely OLE compliant*. Because of these features and the fact that HYSYS is highly customizable, it has been used in this study to simulate both the conventional unit operations such as separators, pumps, compressors, and heat exchangers and other required sub-models such as wells and gathering network.

The combination of Eclipse and its controlling program Open-Eclipse, and the Process Modeling Environment (PME) of HYSYS, therefore make up the mentioned platform. The prototype that was created based on this platform, can justly be called a field simulator since it is capable of simulating the petroleum production system as a whole. This tool or field simulator is a hybrid application in nature and is created by coupling these commercial simulators using technologies such as Automation and Parallel virtual Machine (PVM) interfacing (PVM documentation).

1.3 Integrated system requirements

The inherent complexity in modeling and simulating a petroleum production system is obviously the first hurdle that an integrated simulation tool should overcome. Naturally, application of an integrated system is more complicated as it contains more elements or subsystems. Utilization of an integrated system is therefore viable only if it remains relatively simple to develop and easy to learn and use. Integration of commercially available simulators using currently available inter-process communication protocols is the strategy proposed in this study to develop an integrated system that is not unreasonably difficult to create and is reasonably easy to use.

With no doubts, inclusion of a 3-D reservoir simulator into the integrated system increases the functionality of the system and widens its range of applications. This gain however comes with a price. Heavy calculations of a full-field compositional model, for instance, can reduce the applicability of

* OLE, which stands for Object Linking and Embedding is a technology developed by Microsoft Corporation to allow online data transfer between various applications that run under Windows operating system. Automation, which has evolved from OLE, is the ability to programmatically interact with an application through objects exposed by that application.

the integrated system by increasing total CPU requirements. This problem is specially augmented in iterative calculations, such as optimization studies, where reservoir simulator should be solved a number of times. There are fortunately methods and technologies available to reduce both the reservoir simulation time and number of iterations required by the search routine. An efficient integrated system is accordingly the one that includes or accommodates such methods and technologies.

Process simulator calculations are in general considered not highly CPU intensive. Therefore, integration of process simulation should not cause any special problem. The process simulator is however required to support a communication protocol.

Integration capability and rapid convergence are the basic characteristics expected from the software that simulates wells and gathering network in an integrated system. Finding a commercial simulator for use in the developed integrated system proved to be difficult. An efficient solution proposed in this study is to customize the process simulator to simulate the gathering network. The proposed method is easily integrated in the system and converges reasonably fast.

Field level optimization problems are characterized as multivariate optimization with discrete and rough surfaces. Derivative information is either nonexistent or expensive to obtain. On the other hand, serious limitations exist on total number of iterations. The search algorithm that is used in an integrated field simulator should therefore be able to handle complex search surfaces with minimum number of iterations with no specific requirements regarding the mathematics of the problem in hand. Genetic algorithm search was selected among the few available search algorithms that possess all mentioned characteristics based on reported satisfactory performances in similar problems. Acceptable number of iterations and good computation performance that resulted from application of genetic algorithms in this study justify their use in integrated simulation systems.

Chapter 2

The field simulator

Integrated simulation and open software architecture are probably the way forward leading to the next generation of simulation tools. Using standard interfaces, these techniques will link presently incompatible programs to create more comprehensive simulators. The discussion presented in first chapter explained advantages of integrated simulation. This chapter presents a recipe to create an integrated software in order to demonstrate the simulation power of a fully integrated application created based on open software architecture concept. This chapter is devoted to explaining the methods used in development of the hybrid software that is meant to perform as a petroleum field simulator. This discussion will explain the working principles of the software and clarifies its capabilities and limitations.

2.1 Introduction

The most important elements of a typical petroleum production system are:

- The subsurface hydrocarbon reservoir.
- The production and injection wells.
- The surface chokes.
- The surface production gathering or injection distribution network.
- The processing plant.

This system is in fact a continuous flow system with water and/or gas injection lines acting as recycle streams between the topside facility and the reservoir. An integrated software system is probably one of the best alternatives to model such systems.

The petroleum engineering software industry during its development has adopted the same borders that have existed traditionally between different disciplines within the petroleum industry. This has resulted in development of standalone applications. Three distinct groups of petroleum engineering software that are normally used in field studies are:

- Reservoir simulators.
- Wells and gathering or injection network simulators.
- Process simulators.

The conventional standalone programs that simulate each subsystem are related to each other through boundary conditions. In reservoir simulators, wells and gathering systems are boundary conditions and simple process models are used to simulate surface separation. In wells and network simulators, the boundary conditions are the inflow from the reservoir at one end and the delivery point to the first separator at the other end. In process simulators, a feed stream with specified composition, flow rate, pressure, and temperature at the network delivery point constitutes the boundary condition.

A petroleum production system is in fact a complex of pipe network, recycle streams, and conventional unit operations. Feed to this system is supplied by the hydrocarbon reservoir. Such a system can be modeled and solved using conventional techniques such as tearing and partitioning that are commonly used in process engineering flowsheeting (Biegler, 1997). Using this approach, Nazarian and Golan (2000) showed that it is possible to integrate a simple reservoir model into an existing Process Modeling Environment (PME) to create an integrated petroleum field simulator.

Process engineering software developers are moving towards creating standard interfaces that allow different software components to couple to each other in order to create hybrid simulation tools. The outcome will be that the vendors do not need any longer to produce all required Process Modeling Components (PMC) such as pumps, compressors and separators, and solver routines to solve the flowsheet and at the same time devote enormous resources to develop and maintain the Process Modeling Environment (PME). Instead, there will be a wealthy collection of modeling tools available from various suppliers that user can buy and put together to create his own simulator. This will create far better and more detailed simulation models and saves the user from buying models that he never needs.

Until the time that such approaches are adopted in petroleum industry, one can develop an integrated simulator using one of the following methods:

1. A single program that contains all subsystems and uses common computation engine, common fluid and thermodynamic models and common data storage.
2. A hybrid system formed by linking independent programs that maintain their independent functionalities and do not share computation engines. The link implies transfer of data and request for services.

Although there have been attempts to create a field simulator using the first method (see for example Schiozer, 1994), such approaches have not yet resulted in development of a total field simulator. For the time being, it seems that the tendency in petroleum industry is towards creating hybrid software tools that can function as a field simulator (see press release by Baker Jardine, 2001).

2.2 Background

Several modeling tools have been developed in the past 20 years to model the performance of entire field from the reservoir to the export point. The first generations of these programs were primarily an extension of reservoir simulators to account for the constraints of production network and surface process facilities. Several initiatives have been reported to link commercial simulators to develop a comprehensive total field simulator (Breaux 1984, Hooi 1993, Haugen 1995, Pieters 1995, Linthorst 1997, Deutman 1997, Trick 1998, Weisenborn 2000, and Gayton 2000). Some recent approaches aim at creating a comprehensive field simulator by modeling all required subsystems from basic principles (Schiozer 1994, Bayer 1998, and Bayer 1999).

Hepguler et al. (1997) describe the integration of a commercial reservoir simulator and a commercial production network simulator using the Parallel Virtual Machine interface (PVM documentation). Using the same interface, Trick (1998) reports integration of the black oil reservoir simulator Eclipse to the gas deliverability forecasting model FORGAS (FORGAS documentation, 1997). Mogensen et al. (1998) report a successful application of the method described by Trick to the Sexsmith gas condensate field. Lamey et al. (1999) carried out a dynamic study on the Europa and Mars projects using the transient flow simulator OLGA (OLGA documentation, 2000) coupled with OTISS dynamic process simulator

(OTISS documentation, 2000). Haugen et al. (1995) describe an efficient use of Parallel Virtual Machine interfacing to couple a number of black oil reservoir models to a global production master program that handles the global production and injection constraints.

Schiozer (1994) reports creating an integrated simulation tool by modeling the necessary components of the system implicitly. He develops techniques to couple solutions of performance of reservoir and surface facilities using domain decomposition concept. Bayer et al. (1998) continue Schiozer's work by introducing a preconditioner to reduce the simulation time through variable iteration techniques. In a second paper, Bayer et al. (1999) further their research by proposing an adaptive implicit reservoir formulation and implicit facility coupling to improve the solution technique for the fully coupled model. Beliakova et al. (2000) report development of a tool for rigorous integration of subsurface and surface units. The developed simulator is a collection of simulation models such as flow models, simple gas condensate and black oil PVT models, and EOS flash calculations.

While the attempts aiming at creation of a total simulator using the basic principles seems to be more consistent, such approaches are less likely to create a tool that will have modeling components as powerful as existing commercial softwares that specialize in a specific discipline. It is probably because of this reason that very few attempts have been made in this regard. At the same time, more reported attempts aim at creation of a field simulator using the existing commercial softwares. The recent acquisition of Baker Jardine by Schulumberger can be viewed as an example of such attempts (Baker Jardine, 2001).

Several observations can be made regarding the reported development methods that advocate the approach adopted in this project:

- In-house simulators of major operating companies are complicated, take long time to develop and the development is usually never completed.
- Several in-house simulators of major operating companies used in total field simulators became obsolete with the wave of mergers or by changing policy regarding development and maintenance of in-house developed softwares.

- In-house field simulators are often field specific and are not user friendly. They lack universality and lack the efficiency that is required in multi-client, multi-customer applications.

These deficiencies can be avoided using the approach presented in this project. This study takes the fast route of developing a hybrid application using the available commercial software and demonstrates that this approach results in creation of a powerful and flexible tool that is easy to use in wide range of field studies and requires considerably less effort to develop.

2.3 Available integration techniques

Perhaps the most significant developments in software industry in the mid-90s were object-oriented programming and inter-process communication. These technologies were among the latest products of the need to share data among programs in efficient ways.

Dynamic Data Exchange or DDE is the first standard protocol developed to make automatic inter-process communication possible. This protocol is based on the cut-and-paste metaphor. In a sense, DDE is an automation of the clipboard technology that has been present in Windows from initial release. In another sense, DDE is a method for allowing two applications to converse with one another.

DDE was soon proved to work not that well and therefore was substituted by Microsoft with another technique that is called Object Linking and Embedding or OLE. This technology was made to allow the user to take a particular object such as a spreadsheet and embed it into another object such as a text document. Changes to values in the spreadsheet would automatically be updated in the text document. This was a very powerful feature and was available to users without the added complexity of writing code. It was simply a matter of cutting and pasting the objects.

Automation techniques that later evolved from OLE is presently considered as the state of the art in inter-process communication. Automation is a standard based on Microsoft's Component Object Model (COM) and is the ability to programmatically interact with an application through objects exposed by that application. Automation works in a client-server fashion. A server is the application that provides a service that can be used by clients if they know the proper protocols. Using an Automation client such as Visual

Basic the end user can write the code to access these objects and interact with the Automation server. Access to an application through Automation is language-independent and any application written in languages such as Visual Basic, C++ or Java can interact with the server. HYSYS for example is an Automation server application. By developing the proper code, it is possible to send and receive information to and from HYSYS. The exposed objects make it possible to perform nearly any action that is accomplished through the HYSYS graphical user interface. Appendix A discusses Automation in more details and presents sample Automation codes to clarify various applications of Automation protocols.

Parallel Virtual Machine or PVM interfacing is another technique available to exchange messages between various applications. PVM is a message passing protocol that permits a heterogeneous collection of Unix and/or NT computers linked together in a network to be used as a single large parallel computer. In addition to providing the ability of distributing a single job over a collection of computers and thus solving large computational problems more cost effectively, PVM can programmatically control one application through another just like Automation. PVM has been used by GeoQuest in several of its products for internal or external communication and is the protocol that transfers Open-Eclipse messages to Eclipse simulator. As was discussed earlier, PVM techniques have been used in several attempts to link Eclipse black oil simulator to surface gathering systems.

There are still other available techniques such as pdXi, STEP and OPC that have been used mainly in process engineering software (Braunschweig, 1999).

Although probably not qualifying as an inter-process communication tool, Dynamic Link Library or DLL can be used in such context. A DLL is a software component that can be called from other applications to perform a specific task. Since the use of DLL resembles that of a normal subroutine, it can be used to handle the online data transfer between applications and therefore eliminate the need for time consuming data transfer through print outs.

2.4 The developed tool

Figure 2.1 depicts the components and integration methods that have been proposed in this study to develop the hybrid simulator. The system is composed of three applications; Eclipse reservoir simulator, HYSYS process simulator and LinkControl executive program.

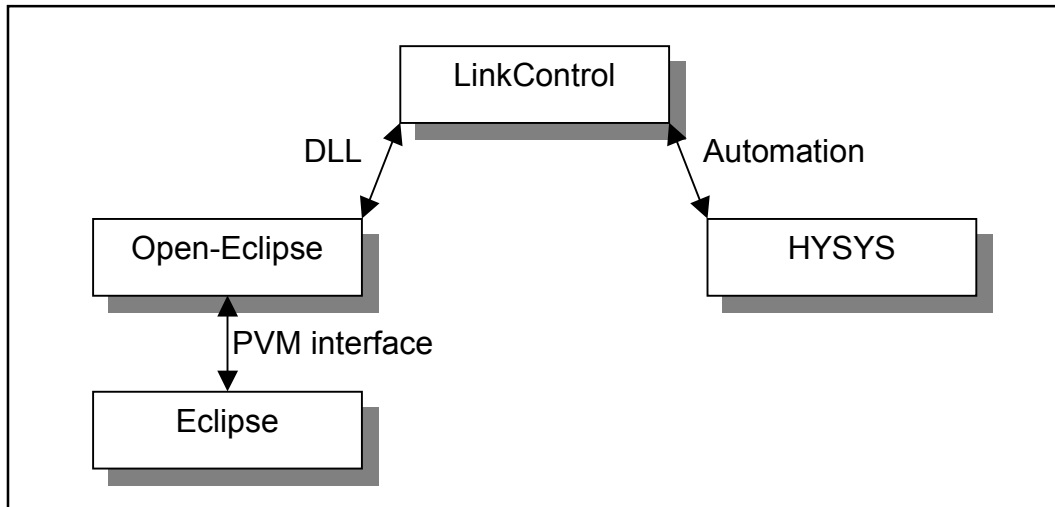


FIGURE 2.1 Integrating reservoir and process simulators using a combination of Automation techniques and PVM interfacing.

LinkControl was developed for the purpose of this study using Visual Basic language to provide the required user interface, to act as the system executive and to perform the required translation and interpretation of the results of the other two programs. The interfaces that user is required to interact with during simulation are those provided by LinkControl.

HYSYS compositional process simulator has a central role in this system and performs the job of modeling a considerable number of system components. Its open software structure allows complete customization to particular needs of the system. In the developed system, HYSYS is not just a process simulator. It acts in fact as a programming environment in which most of the system has been constructed.

Simulation in HYSYS normally begins by creating a Property Package that is composed of a thermodynamic model and a number of components. The

thermodynamic model can be a built-in Equation of State or an activity model. The required components can either be selected from a comprehensive list of library components or can be created as hypothetical components by supplying the component's basic physical and critical properties. Built-in correlations are used to estimate the rest of required properties using the supplied basic property data. The HYSYS Oil Manager can be used to convert available lab results such as ASTM or TBP tests to a fluid characterization with the required number of components.

In the developed system, HYSYS acts in four basic capacities; 1) it simulates the topside facilities using its own built-in models, 2) it simulates production and injection wells by performing rigorous multiphase pressure and temperature calculations, 3) it simulates the production gathering network using its pipe model and logical operations, and 4) it is used as a calculation engine for phase or component property calculation whenever required.

Communication between LinkControl and HYSYS is through Automation protocols. LinkControl can load and link existing simulation cases in HYSYS or use HYSYS calculation engine in various sub-models when necessary.

Eclipse simulator is used to model the reservoir behavior. As was mentioned before, Eclipse is by itself a batch program controlled by the contents of its data file. Therefore, in order to control it programmatically inside a dynamically linked system, Open-Eclipse protocols have to be used. The communication between Eclipse and Open-Eclipse is through Parallel Virtual Machine interface. The Open-Eclipse controlling program has to be developed using FORTRAN language. The only way to communicate programmatically between the LinkControl's Visual Basic code and the Open-Eclipse's FORTRAN code is to compile the Open-Eclipse code into a Dynamic Link Library and call it through the Visual Basic code. This way the communication between Eclipse simulator and the rest of the system becomes dynamic and an integrated system is developed. The information that Open-Eclipse normally receives from the keyboard can then be sent through the DLL. The simulation results, which are normally printed to a file, are then passed online to LinkControl through the DLL for interpretation and further transmission to HYSYS.

No manual intervention is required during simulation using the developed tool, however the HYSYS simulation cases and the Eclipse input file should be prepared by the user before beginning the simulation study using the field simulator. There are several important points that user must bear in mind while preparing HYSYS cases or Eclipse input file. For example, the number of wells used in the Eclipse file should equal the number of inflow streams in HYSYS gathering network simulation case. These practical issues have been addressed in Appendix B. **Figure 2.2** clarifies the difference between manual and dynamic links in the developed tool.

The simulation normally begins by loading existing HYSYS cases that simulate gathering network and processing unit, separately. Because of iterative nature of calculation in gathering network simulation, it was decided to simulate gathering network and processing unit by the use of two separate HYSYS cases in order to avoid unnecessary iteration in the solution of the processing unit simulation case. After the cases are loaded, the user selects the Eclipse file that has already been prepared and saved in the system. The user then decides on how the reservoir simulator should proceed. There are two options available to run the Eclipse simulator by the commands sent through the developed Open-Eclipse master program. It is possible to run Eclipse either for one time-step at a time or for a definite number of time-steps. After gathering the required information, LinkControl starts Eclipse simulation and waits for the return message from Open-Eclipse. When the return message is received, LinkControl will query the required data from Eclipse. The data is then interpreted and modified by LinkControl and is sent to HYSYS simulator to perform the required calculations.

HYSYS results are then used depending on the study requirements. These results for example can be used to calculate objective function for optimization purposes or can be used to calculate the net present value for a production scenario.

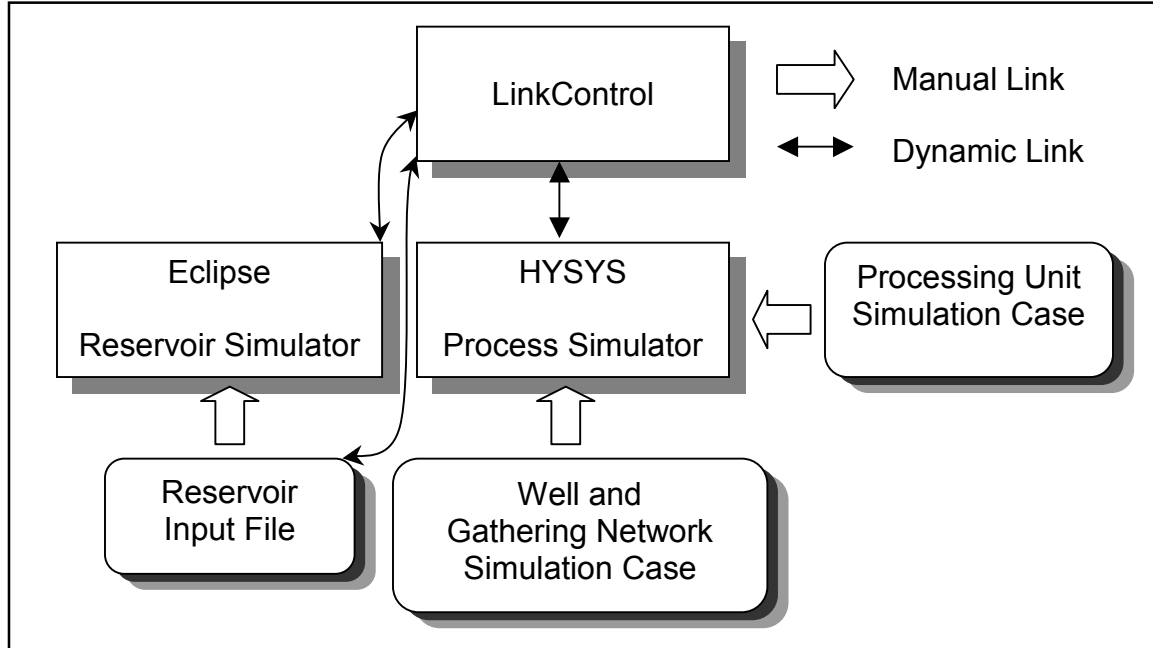


FIGURE 2.2 Interaction between components of the hybrid application.

In case of water and/or gas injection into the reservoir, the specifications of these injection streams are determined by HYSYS. The information regarding injection streams is sent back to Eclipse through LinkControl and is used in the next time-step. **Figure 2.3** depicts a more detailed view of the tool and the transferred data.

The developed Open-Eclipse controlling program is capable of spawning and controlling both Eclipse 100 black oil and Eclipse 300 compositional simulations. The use of Eclipse 300 in parts of this study is because of the fact that HYSYS is a compositional simulator and compositional data from Eclipse can be easily transferred to HYSYS.

The data inquired by LinkControl from Eclipse 300 are the well stream hydrocarbon composition, Z_{HC} , bottomhole flowing pressures, p_{wf} , well stream hydrocarbon molar flow rate, m_{HC} , and well stream water molar production rate, m_w . These data are modified in LinkControl to create the equivalent units required in HYSYS. Water is treated as a phase in Eclipse while is considered as a component in HYSYS. LinkControl accordingly calculates a new composition with water as a component using the data from Eclipse and transfers the modified data to HYSYS as total composition Z_i , and total molar flow, m_{total} .

The HYSYS simulator that now simulates the rest of the system transfers these data to the well, choke, network and process models, respectively and solves all streams and unit operations. From HYSYS calculation results, the rest of the required parameters are calculated by LinkControl depending on the type of study.

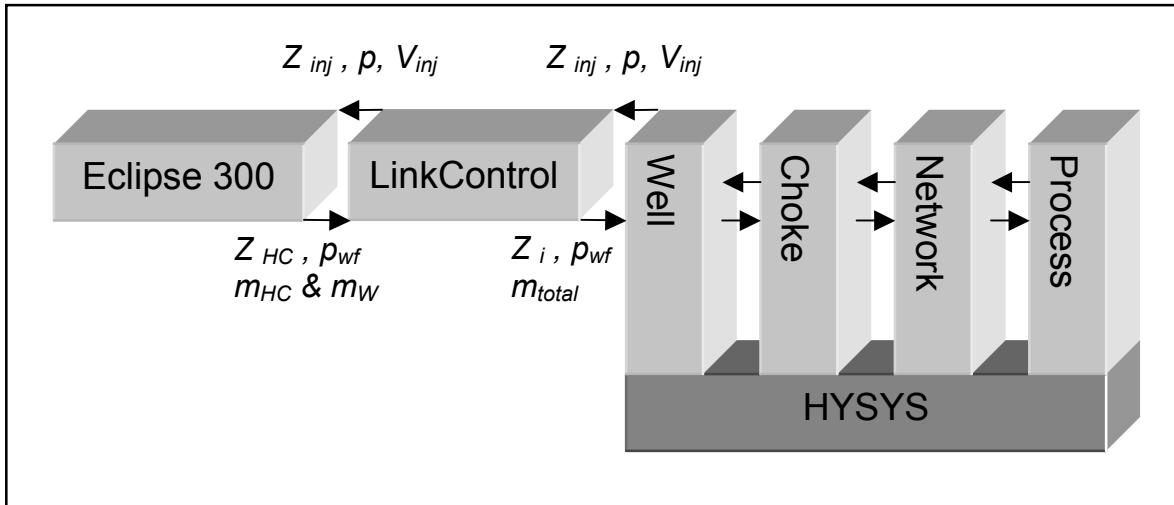


FIGURE 2.3 Data transfer between components of the system.

The information related to injection streams is used by LinkControl to modify the Eclipse 300 input file in order to include the pressure, injection rate and, in case of gas injection, the composition of the injected stream into the Eclipse 300 input file.

Although the choices for reservoir and the processing unit simulators were almost clear, the choice of modeling tool for wells and gathering network was not that obvious. Eclipse and HYSYS were obvious choices because of the reasons mentioned earlier but there were several alternatives to model wells and gathering network.

For modeling the wells and gathering network, one option was to use the transient flow simulator OLGA. This simulator was not used because of steady state nature of this study. On the other hand, OLGA was not easily available for integration into the rest of the system in the way it was required. The performance of OLGA in solving compositional gathering networks with several branches was not acceptable either. Very good initial assumptions were required to begin the solution and nearly complete PVT

data that cover the possible ranges of pressure and temperature had to be developed prior to performing the simulation. Finally, the use of third software could make the whole system more complicated to learn and use.

The other option to simulate the wells was to use the well flow models available in GeoQuest package. These modeling facilities in GeoQuest were not available through Open-Eclipse, making it next to impossible to communicate with these applications dynamically.

Although HYSYS is not a piping and hydraulic simulator, through its wealthy collection of models and solvers, it supplies enough tools to solve multiphase compositional gathering networks in an efficient way as will be discussed in the coming sections. The Pipe Flow unit operation in HYSYS performs rigorous multiphase pressure and temperature calculation and therefore can be used to simulate multiphase compositional flow in the wells. In solving a compositional system, HYSYS does not require any preprocessors to generate the required PVT data. The required properties are calculated automatically and are stored in a place where all unit operations can use. Because of these capabilities, HYSYS was chosen to simulate the wells and gathering network in this study. The use of HYSYS to simulate well flow has been explained in section 2.5 and the pros and cons of using HYSYS to solve the gathering network have been discussed in section 2.6.

The product of the integration technique that was discussed in this section is a flexible hybrid program that is capable of simulating the whole petroleum production system using only Eclipse and HYSYS simulators. The possibility of doing Automation with this hybrid simulator widens the range of applications of this system. The optimization studies that are carried out in the next chapter are examples of how the open software structure of the developed system enables integration of other software into the system to perform the required tasks.

2.5 Well model

The multiphase compositional flow in the wells has been modeled in this study using Pipe Flow unit operation in HYSYS. The Pipe Flow unit operation can be used to simulate a wide variety of piping situations ranging from single or multiphase plant piping with rigorous heat transfer estimation, to large capacity looped pipeline problems. It offers two pressure drop correlations; one developed by Gregory, Aziz, and Mandhane (1975), and

the other by Beggs and Brill (1973). A third option, OLGAS, is also available as a gradient method. Four levels of complexity in heat transfer estimation are allowed in order to find a solution as rigorous as required while at the same time allowing for quick solution to well-known problems. The Pipe Flow unit operation offers three calculation modes; pressure drop, flow, and pipe length. The appropriate mode will automatically be selected depending on the information supplied. In order to solve the pipe, enough information must be supplied to completely define both the material and energy balances.

The Pipe can solve in either direction. The solution procedure generally starts at the end where the temperature is known. Temperature is typically not known on both ends. HYSYS then begins stepping through the pipe from that point, using either the supplied pressure, or estimating a starting value. If the starting point is the pipe outlet, HYSYS steps backward through the pipe. At the other end of the pipe, HYSYS compares the calculated solution to other known information and specifications, and if necessary, restarts the procedure with a new set of starting estimates.

There are two different methods for calculating the pressure drop:

Method 1: If the temperature and pressure have been specified at the same end of the pipe, then energy and mass balances are solved for each increment and the temperature and pressure of the stream at the opposite end of the pipe are determined.

Method 2: If temperature for one stream and pressure for the other have been supplied, the following iterative loop is used outside of the normal calculation procedure:

- A pressure is estimated for the stream that has the temperature specified.
- The pressure and temperature for the stream at the opposite ends of the pipe are determined from incremental energy and mass balances as in the first method.
- If the calculated pressure and user-specified pressure are not within a certain tolerance, a new pressure is estimated and the incremental energy and mass balances are solved again. This continues until the absolute difference of the calculated and user-specified pressures are less than a certain tolerance.

2.5.1 Beggs and Brill pressure gradient

The Beggs and Brill method is based on the work done with air-water mixture at many different conditions and is applicable to inclined flow. In the Beggs and Brill correlation, the flow regime is determined using the Froude number and inlet liquid content. The flow map used is based on horizontal flow and as depicted on **Figure 2.4** consists of four flow patterns; segregated, intermittent, distributed and transition. Once the flow regime has been determined, the liquid holdup for a horizontal pipe is calculated, using the correlation applicable to that regime. A factor is applied to this holdup to account for pipe inclination. From the holdup, a two-phase friction factor is calculated and the pressure gradient is determined.

2.5.2 Gregory Aziz Mandhane pressure gradient

For the Gregory Aziz Mandhane correlation, an appropriate model is used according to **Table 2.1** for predicting the overall pressure drop in two-phase flow. The flow regime map for this method is shown on **Figure 2.5**.

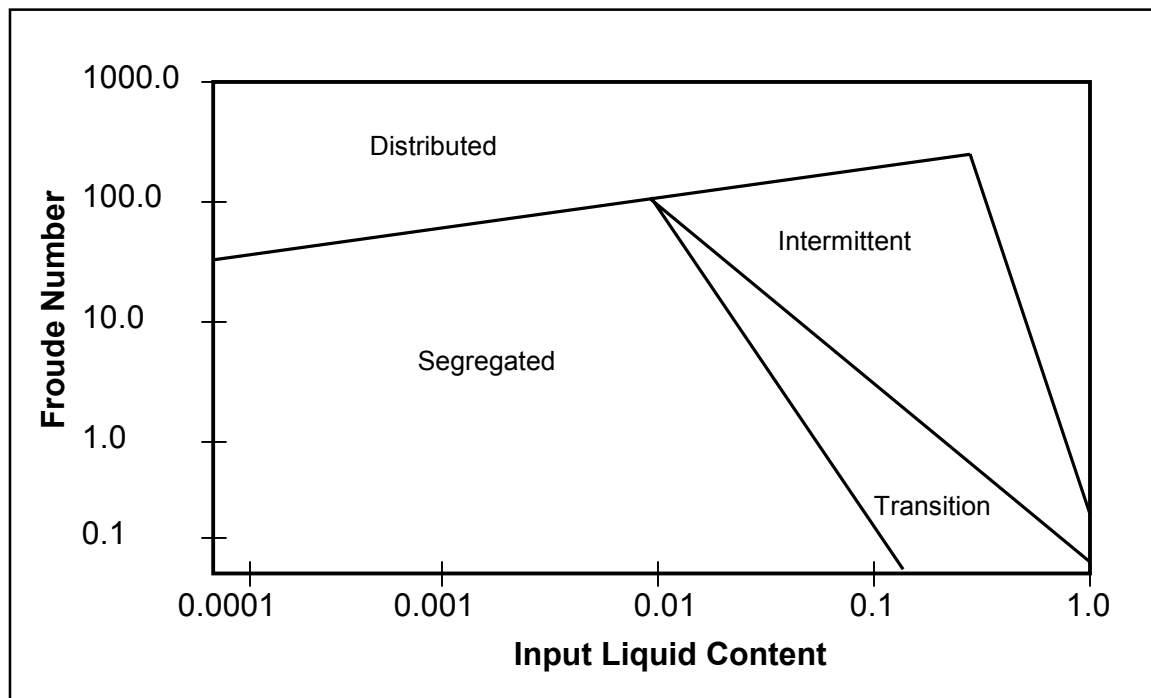


FIGURE 2.4 Beggs and Brill flow regimes.

2.5.3 Two-Fluid model (OLGAS) pressure gradient

The two-fluid model (OLGAS) employs mechanistic models for each of the four major flow regimes; stratified, annular, slug and dispersed bubble flow. It predicts the pressure gradient, liquid holdup and flow regime. It has been tested in one-degree increments for all angles from horizontal to vertical. OLGAS gives one of the best overall predictions of pressure drop and liquid holdup of any currently available method.

All methods account for static head losses, while only the Beggs and Brill and OLGAS methods account for hydrostatic recovery. Beggs and Brill calculate the hydrostatic recovery as a function of the flow parameters and pipe angle.

TABLE 2.1 Calculation models for various flow regimes in Gregory Aziz Mandhane method.

Regime	Model
Slug Flow	Mandhane modification no.1 of Lockhart-Martinelli
Dispersed Bubble	Mandhane modification no.2 of Lockhart-Martinelli
Annular Mist	Lockhart-Martinelli
Elongated Bubble	Mandhane modification no.1 of Lockhart-Martinelli
Stratified	Lockhart-Martinelli
Wave	Lockhart-Martinelli

The performance of pipe segment unit operation has been demonstrated in **Example 2.1** in connection with the solution to gathering networks.

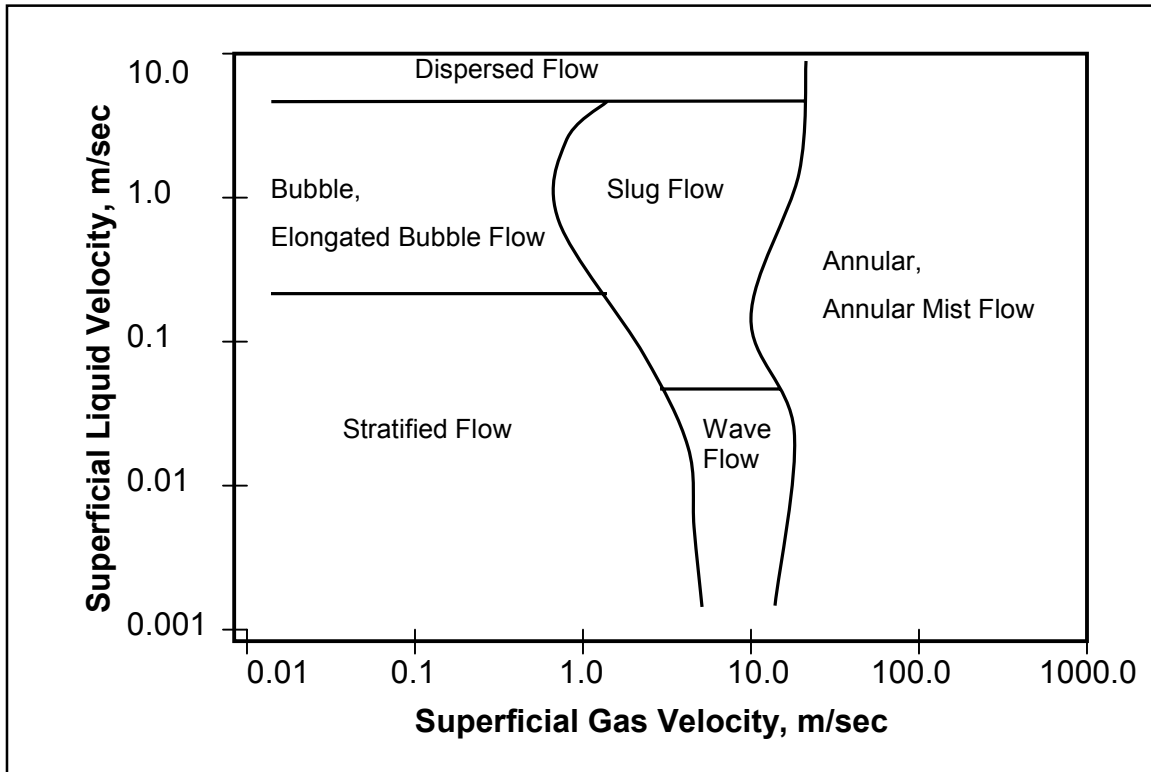


FIGURE 2.5 Gregory Aziz Mandhane flow regimes.

2.6 Gathering network model

From the hydrodynamics point of view, the hydrocarbon production piping network falls into the category of compressible, multiphase, multi-component network systems. Simulation of such networks and finding robust solution algorithms that converge after a reasonable number of iterations is far from being trivial.

The literature on single-phase flow network is quite rich and there are a number of methods available for simulating such networks (see for example, Bender 1979, Isaacs 1980, Wood 1981, Dupuis 1987, and Haghghi 1992). Despite the fact that an intense research activity has been devoted to modeling the compressible, multiphase, multi-component pipe flow, there are few reported studies that address modeling the networks of such flow systems.

The pipe networks can roughly be classified into three major groups; converging networks, diverging networks and looped networks. From the

viewpoint of network solution techniques, converging networks are the easiest to solve. The solution becomes more complicated when loops and diverging branches exist in the network. Presence of loops creates non-linearity in the system and makes the solution more complicated. The problem with diverging networks is that at present there is no solid theoretical basis to model the diverging multi-component tees. At a tee element, where a multi-component stream is split into two or more branches, the way each component is divided between diverging streams depends not only on its equilibrium properties but on its hydrodynamic properties too. Therefore, depending on size, shape and elevation change of the tee element, very different situations may arise and the components may be distributed in many different ways.

Most of the well-known steady state or transient multiphase flow simulators perform quite poor when it comes to simulation of pipe networks. They normally require very good initial estimates of the solution and fail to capture truly the multiphase equilibrium that is established between the phases at various points along the network. Using such tools, the simulation of pipe loops or diverging networks is next to impossible.

2.6.1 Introducing the HYSYS.process as a network solver

As was explained before, HYSYS is an integrated, fully compositional flowsheeting tool normally used to simulate chemical and petrochemical processes. In addition to numerous built-in thermodynamic routines and databanks, HYSYS provides a wealthy collection of unit and logical operations. Using Pipe Flow, Mixer and Tee unit operations together with Adjust logical operation, it is possible to simulate and solve complex flow networks with reasonable accuracy and speed. The powerful and user-friendly Process Modeling Environment (PME) of HYSYS allows rapid design of the network Process Flow Diagram (PFD). The solution of the network is highly facilitated since the sophisticated PME of HYSYS discovers the network topology using standard tearing and partitioning procedures and finds suitable algorithms to solve it very efficiently. After designing the network and defining the problem boundaries, one can exploit the power of logical operations within HYSYS to set up a search algorithm to converge to the solution of the flow network.

As was discussed in section 2.5, HYSYS provides a multiphase, compositional pipe flow model that can be used readily in a network model

to simulate the pipe flow. Mixer and Tee unit operations available in HYSYS can simulate the situations where flow is received or distributed by a node, respectively, making it possible to model network nodes in HYSYS. Finally, using nested simultaneous Adjust logical operations, a search method can be set up to converge to the solution of the pipe network.

The applicability of this method to converging networks has been demonstrated here. Nazarian (2000) has discussed the applicability of this method to diverging and looped networks. Here more emphasis has been put into solving large converging networks as these types of networks are commonly used in the petroleum production gathering systems.

In the sections that follow, the required HYSYS unit operations to model and solve a typical gathering network have been presented and discussed in some detail. **Example 2.1** clarifies the subject by showing the solution to a gathering network problem.

2.6.2 Mixer operation

The Mixer operation combines two or more inlet streams to produce a single outlet stream. A complete heat and material balance is performed with the Mixer. This means that the one unknown temperature among the inlet and outlet streams will always be calculated rigorously. If the properties of all inlet streams to the Mixer that is, temperature, pressure and composition are known, the properties of the outlet stream will be calculated automatically since the composition, pressure and enthalpy will be known for that stream.

The mixture pressure and temperature are usually the unknowns to be determined. However, the Mixer will also calculate backwards and determine the missing temperature for one of the inlet streams if the outlet is completely defined. In this latter case, the pressure must be known for all streams.

The Mixer will flash the outlet stream using the combined enthalpy. When the inlet streams are completely known, no additional information needs to be specified for the outlet stream. The problem is completely defined since no degrees of freedom remain. The resultant temperature of the mixed streams may be quite different from those of the feed streams due to mixing effects.

2.6.3 Adjust logical operation

In a flowsheet, a certain combination of specifications may be required which cannot be solved directly. These types of problems must be solved using trial-and-error techniques. To quickly solve flowsheet problems that fall into this category, the Adjust operation can be used to automatically conduct the trial-and-error iterations.

The Adjust operation will vary the numerical value of a variable (the independent variable) belonging to a selected stream, in order to meet a required value or specification (the dependent variable) in another stream or operation.

The Adjust operation can be used to solve for the desired value of just a single dependent variable, or multiple Adjusts can be installed to solve for the desired values of several variables simultaneously.

The Adjust can perform the following functions:

- Adjust the independent variable until the dependent variable meets the Target Value.
- Adjust the independent variable until the dependent variable equals the value of the same variable for another Object, plus an optional offset.

Adjust loops can be solved either individually or simultaneously. If the loop is solved individually, the choices are either a slow and sure Secant or a fast but not as reliable Broyden search algorithm. The simultaneous solution method uses a multivariable Broyden search algorithm. A single adjust loop can be solved in the simultaneous mode, however, this method is usually reserved for multiple inter-linked loops.

2.6.4 Solution of converging pipe networks

In most cases, the petroleum production gathering system falls into category of compressible, multiphase, multi-component, converging networks. In such pipe systems, different flow streams with different pressures, temperatures and compositions commingle together to form usually a single outlet stream. To simulate such a system, a Pipe unit operation can be used to model the flow in pipe segments and a Mixer unit operation can model the

converging junctions. In a network problem, usually all required data regarding size, material and topology of piping system, surrounding temperature and type, as well as composition of entering streams are known. The problem is then initialized by defining the system boundaries, which can be either a pressure or a flow boundary. The choice of boundary depends mainly on type of the problem.

An alternative solution is to find the pipe sizes that create physically possible internal rates and pressures, so that both pressure and flow rate boundaries are satisfied.

To begin the network calculation, temperature, pressure, composition, and flow rate of entering streams should be supplied. The type of material from which surrounding is made plus surrounding temperature should be specified as well for energy balance calculation. HYSYS then calculates flow rates, pressures and compositions for outlet and all internal streams. HYSYS can converge to a physically meaningful solution using an iterative procedure that can be built into the simulation using the Adjust logical operation. The following example demonstrates the solution of a converging network using HYSYS.

EXAMPLE 2.1 *Solution of a converging, multi-component, multiphase network.*

The PFD of a converging, multi-component, multiphase network simulated by HYSYS has been depicted in **Figure 2.6**. As shown on this figure, five inflow streams enter the network and commingle to form a single outlet stream. The composition, molar flow rate and pressure values required to define these entering streams are directly inquired from the reservoir simulator. The temperature value is manually entered both in the reservoir input data and in the HYSYS simulation case.

As was mentioned in section 2.5, multiphase, multi-component flow in the wells can be modeled by the Pipe unit operation in HYSYS. In this example, the existing five wells have been modeled using multi-segment, vertical pipes with complete data regarding the insulation and surrounding type and temperature in order to perform a rigorous simultaneous pressure drop and heat loss calculation along the pipe. A linear thermal gradient has been assumed in the ground. The thermal effects of annular fluid such as

convection heat transfer cannot be considered since a pipe-in-pipe heat transfer system cannot be modeled using Pipe unit operation.

Steady state heat transfer is assumed in the ground and the transient heat transfer has been neglected. However, the variations of overall heat transfer coefficient with depth have been considered in calculations.

The pipe segments in the gathering network have been modeled using Pipe unit operation as well. In this example, the pipelines are considered to be unburied, insulated, and horizontal. However, other possible pipe trajectories, insulation types and buried pipes can easily be modeled. By defining the surrounding type and temperature, a rigorous pressure and temperature calculation can be set up.

The five Adjust logical operations together with five Valve operations that model the chokes in the system set up the iterative method that converges to the solution of the network. For example, Adjust operation ADJ-1 changes the pressure drop in Valve VLV-102 in order to equalize pressures of streams Out-2 and Out-1. All other Adjust operations perform the same iterative calculation simultaneously in order to get the same pressure at existing pipe junctions. These junctions are simulated using Mixer unit operations. The final Adjust operation, i.e., ADJ-5 changes the pressure drop of the Valve VLV-101 in order to set the pressure of the Outlet stream at the desired value. By doing so, ADJ-5 sets up the other boundary of the network and the problem is then completely defined. After the simultaneous calculations of all Adjust operations are finished, the system converges to a situation where there is no pressure discontinuity in the network. This means that the same pressure will be determined for streams Out-3, Out-4, Out-5, Out-6, Out-7, and Outlet since there is no source of pressure drop between these streams. Appendix D presents detailed simulation results for **Example 2.1**.

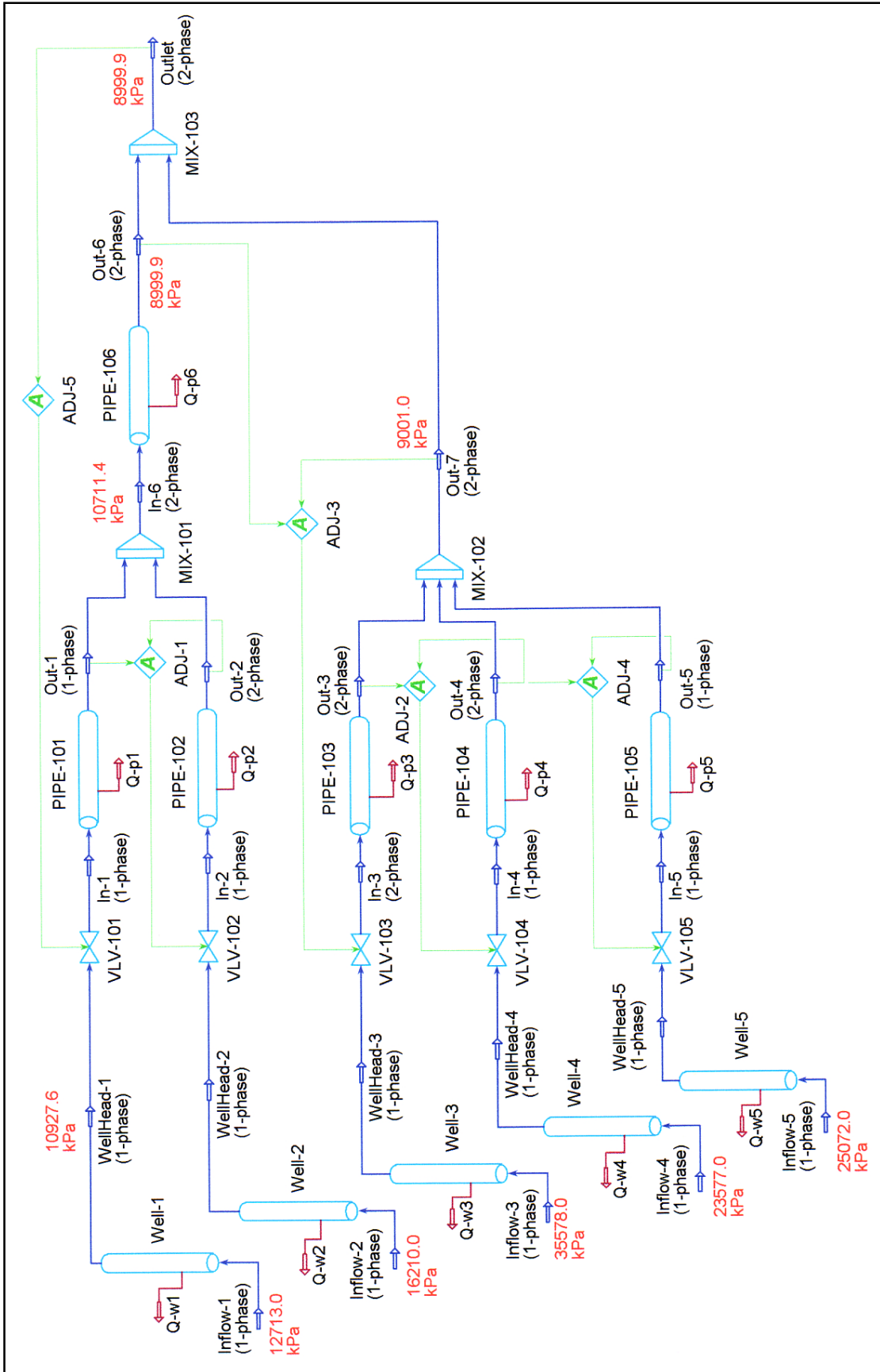


FIGURE 2.6 Process flow diagram of the converging, multiphase flow network presented in Example 2.1.

For solution of a gathering network of the type discussed in this example, performance of HYSYS is quite acceptable. The initial iteration may take few minutes but in subsequent simulations where gradual change of the input data from the reservoir is introduced at the network entrance, the convergence is much faster. The simulation using this method does not involve application of any preprocessor to generate the required PVT data. No initial assumptions regarding the solution are required either. This will highly speed up the task of modeling compared to simulators such as OLGA that need a considerable preparation work before actual simulation can begin.

The composition used in this example for inflow streams creates flow situations that range from complete gas flow to two-phase flow and to complete liquid flow. Stream Inflow-2 introduces water to the system creating a multiphase flow situation in parts of the network. Despite all these complexities, the property calculation routines function satisfactory and the method manages to solve this multiphase, multi-component system in an efficient way.

As was briefly mentioned before, this method is mostly suitable for converging pipe networks. In diverging networks, where streams split into two or more branches, the user should decide the percentage of flow that will be distributed between each diverging stream. Obviously this is not a realistic situation where pressure difference dictates the size of the flow. The problem of diverging branches and Tee operations are solved in the dynamic simulator HYSYS.plant where pressure difference dictates the size and direction of flow. Unfortunately, for the time being Pipe unit operation is not available in dynamic mode. Therefore, this method does not seem to capture the actual behavior of diverging and looped network systems. Fortunately, unlike gas distribution networks, most of the existing petroleum gathering networks are converging pipe networks and the presented method can be used to model their behavior.

2.7 Reconciliation of Eclipse and HYSYS simulations

One of the most important issues in hybrid applications like the one developed in this study is the applicability of the results obtained from one simulator in other participating simulators. Several important facts should be carefully observed before the output of an application can be transferred for use in another application. Neglecting these important points can create

completely wrong results from the hybrid application even if the participating programs perform correct simulation.

In the following section, each of these important issues has been discussed and practical methods are presented to translate the results between various applications participating in the developed system.

2.7.1 EOS tuning in Eclipse and HYSYS

Eclipse 300 and HYSYS are both compositional simulators and use thermodynamic models to calculate the required PVT properties of the present fluid phases. HYSYS has a much bigger collection of thermodynamic models that include nearly all available Equations of State plus many activity models. Eclipse only supports Peng-Robinson, Redlich-Kwong, Soave-Redlich-Kwong and Zudkevitch-Joffe-Redlich-Kwong Equations of State (Eclipse reference manual, 2002).

Although HYSYS supports all the Equations of State (EOS) that Eclipse supports, the actual formulations that are used for each EOS in HYSYS are quite different from that of Eclipse. Accordingly, the EOS calculation results of Eclipse and HYSYS for the same components even with the same physical properties are very different. In fact, in some cases the calculation results can deviate from each other by three orders of magnitude. Therefore, a method is required to tune Equations of State in Eclipse and HYSYS in order to obtain EOS calculation results as similar as possible from these simulators.

The basic problem in tuning EOS formulations available in HYSYS is the fact that HYSYS does not allow modification of any part of the EOS and the only parameter available for tuning is basically the Binary Interaction Parameters (BIP). Modification of these parameters may help producing similar K-values in HYSYS but density values will probably never match since the volume translation parameters are not available in any of the EOS formulations that HYSYS provides. Therefore, it is practically impossible to reproduce the liquid volume calculations using the built-in EOS formulations in HYSYS.

Fortunately, HYSYS provides a Generalized Cubic Equation of State (GCEOS) that makes it possible to include many of the required parameters to reproduce the calculation results obtained from Eclipse EOS calculations.

Peng-Robinson EOS (PR) is perhaps the most widely used EOS in petroleum engineering calculations and is available in Eclipse for complete tuning. In this study, PR has been selected as the base EOS for use in Eclipse simulator. A method is accordingly proposed to tune a generalized cubic EOS in HYSYS to reproduce results obtained from Eclipse EOS calculations using PR. Although the tuning method is discussed in connection with PR, it is basically applicable to all other available Equations of State.

2.7.2 The Cubic Equation of State

The generalized Cubic Equation of State that has been formulated in HYSYS can be presented by the following relation (Reid, Prausnitz, Poling, 1987):

$$p = \frac{RT}{v - b} - \frac{a(T)}{v^2 + ubv + wb^2} \quad 2.1$$

An equivalent form of **Equation 2.1** is:

$$Z^3 + C_1Z^2 + C_2Z + C_3 = 0 \quad 2.2$$

where

$$\begin{aligned} C_1 &= Bu - B - 1 \\ C_2 &= B^2w - B^2u - Bu + A \\ C_3 &= -(B^3w + B^2w + AB) \end{aligned} \quad 2.3$$

$$Z = \frac{pV}{RT} \quad 2.4$$

$$A = \frac{a_{mix}p}{R^2T^2} \quad 2.5$$

$$B = \frac{b_{mix}p}{RT} \quad 2.6$$

a_{mix} and b_{mix} are defined using the following relations in which MR_{ij} is the applied mixing rule:

$$a_{mix} = \sum \sum x_i x_j \sqrt{a_i(T) a_j(T)} \times MR_{ij} \quad 2.7$$

$$b_{mix} = \sum x_i b_i \quad 2.8$$

The temperature dependent parameter a_i is determined as:

$$a_i(T) = a_c \alpha \quad 2.9$$

where

$$a_c = \left(\frac{3 + (u - w)\xi^2}{3 + (u - 1)\xi} + u\xi \right) RT_c V_c \quad 2.10$$

and parameter b_i is given by:

$$b_i = \xi V_c \quad 2.11$$

where

$$[u(w + u) - w]\xi^3 + 3(w + u)\xi^2 + 3\xi - 1 = 0 \quad 2.12$$

Values of b_i and a_c can be determined after determination of ξ using **Equation 2.12**. In HYSYS, the values of u and w can be decided by the user through selecting one of the built-in Equations of State

The value of a_i in **Equation 2.9** can be determined by calculating α using the following relation:

$$\alpha(T) = \left[1 + \kappa(1 - T_R^{0.5}) \right]^2 \quad 2.13$$

The κ parameter in **Equation 2.13** is calculated using a polynomial equation containing six κ parameters as shown below:

$$\kappa = \kappa_0 + \left(\kappa_1 + (\kappa_2 - \kappa_3 T_R)(1 - T_R^{\kappa_4}) \right) \times (1 + T_R^{0.5})(0.7 - T_R) \times T^{\kappa_5} \quad 2.14$$

κ_0 is calculated by the following relation in which ω is the component acentric factor:

$$\kappa_0 = A + B\omega + C\omega^2 + D\omega^3 \quad 2.15$$

Appropriate numerical values of κ_1 to κ_5 are entered in HYSYS to calculate value of parameter κ . The values of A , B , C , and D are entered according to the corresponding built-in Equation of State that is used to initialize the GCEOS.

The GCEOS in HYSYS considers volume translation correction to provide a better calculation of liquid volume by the Cubic Equations of State. The volume shift correction provides a translation along the volume axis that results in a better calculation of liquid volume without affecting the vapor liquid equilibrium (VLE) calculations. The volume shift implemented in HYSYS can be presented as:

$$\tilde{v} = v - \sum_{i=1}^n x_i c_i \quad 2.16$$

$$\tilde{b} = b - \sum_{i=1}^n x_i c_i \quad 2.17$$

where \tilde{v} is the translated volume, \tilde{b} is the translated cubic equation of state parameter, c_i is the pure component translated volume and x_i is the mole fraction of component i in the liquid phase.

The resulting equation of state will appear as shown in **Equations 2.4**, and **2.6** with v and b replaced with the translated values \tilde{v} and \tilde{b} , respectively.

2.7.3 The Peng-Robinson Equation of State

The 1979 version of Peng-Robinson Equation of State that is used in Eclipse is given by the following relation (Whitson, 1994):

$$p = \frac{RT}{v - b} - \frac{a}{v(v + b) + b(v - b)} \quad 2.18$$

or in equivalent form in terms of Z-factor:

$$Z^3 - (1 - B)Z^2 + (A - 3B^2 - 2B)Z - (AB - B^2 - B^3) = 0 \quad 2.19$$

The constants of **Equation 2.18** are given by:

$$a = \frac{R^2 T_c^2}{p_c} \times \alpha \times \Omega_a^o \quad \Omega_a^o = 0.457235529 \quad 2.20$$

$$b = \frac{RT_c}{p_c} \times \Omega_b^o \quad \Omega_b^o = 0.07796074 \quad 2.21$$

$$\alpha = [1 + m(1 - \sqrt{T_R})]^2 \quad 2.22$$

where

$$m = 0.3796 + 1.485\omega - 0.1644\omega^2 + 0.01667\omega^3 \quad 2.23$$

The volume translation that has been implemented in Eclipse is expressed as:

$$v_L = v_L^{EOS} - \sum_{i=1}^n x_i c_i \quad 2.24$$

$$v_V = v_V^{EOS} - \sum_{i=1}^n y_i c_i \quad 2.25$$

where v_L^{EOS} and v_v^{EOS} are EOS-calculated volumes, x_i and y_i are liquid and vapor compositions and c_i are the component-dependent volume shift parameters.

2.7.4 The proposed tuning method

In Peng-Robinson EOS, it is common to consider two parameters C_a and C_b in **Equations 2.20** and **2.21**, as shown in the following modified equations:

$$a = \Omega_a^o \frac{R^2 T_c^2}{p_c} \times \alpha \times C_a \quad 2.26$$

$$b = \Omega_b^o \frac{RT_c}{p_c} \times C_b \quad 2.27$$

The use of these two parameters is quite common and many PVT simulators that generate compositional input for use in compositional reservoir simulators include these two parameters as part of the compositional data.

Eclipse 300 implementation of PR considers the effects of these two parameters, but unfortunately, in current version of the generalized Cubic Equation of State in HYSYS these parameters have not been implemented. Because of this limitation, it was discovered in this study that it was not possible to obtain similar results from HYSYS and Eclipse EOS calculations in cases where the effects of these two parameters have been included to create the required reservoir PVT model.

A method suggested by Whitson (2001) solves this problem through generating a new set of T_c , p_c and ω for each participating component by regressing on the values of EOS parameters a and b calculated using a PVT simulator.

The following steps summarize the method suggested by Whitson:

1. For temperature values from T_{min} to T_{max} calculate parameter a , both by considering the effect of C_a , i.e., $a_{C_a \neq 1}$ using a PVT simulator, and as $a = f(C_a = 1, T_c', p_c', \omega')$.

2. Calculate $b_{C_b \neq 1}$ using a PVT simulator and $b = f(C_b = 1, T_c', p_c')$.
3. Find T_c', p_c', ω' by minimizing

$$\sum \left((a^{PVT} - a^{GCEOS})^2 + (b^{PVT} - b^{GCEOS})^2 \right)$$

a^{PVT} and b^{PVT} indicate values calculated using a PVT simulator while a^{GCEOS} and b^{GCEOS} indicate values calculated by the Generalized Cubic Equation of State.

This procedure was tested using PVTx simulator (Whitson, 1995) to generate the required PVT data. Solver program in Excel spreadsheet was used to perform the required regression.

Tables 2.2 indicates the composition used in the test calculation together with component properties calculated by PVTx. **Table 2.3** shows the properties that have been modified using the proposed regression method.

To compare the EOS calculation results from Eclipse and HYSYS using the suggested method, a Property Package was defined in HYSYS using the GCEOS and hypothetical components created using the modified properties generated by the proposed regression method. A simple flash calculation is performed in HYSYS at given pressure and temperature using a separator unit operation in order to determine the gas and liquid phase compositions and phase properties. To perform the same calculation in Eclipse, a single block reservoir model was defined at the same pressure and temperature that was used in the separator unit operation in HYSYS. The initial composition in the block was defined to be the same composition that was flashed in the separator. The compositional data used in Eclipse input file was generated using PVTx simulator.

The calculation results obtained from Eclipse and HYSYS using the mentioned method are shown in **Tables 2.4 to 2.6**. The HYSYS GCEOS calculation result is greatly improved upon using the new set of T_c, p_c and ω generated for each pseudo component defined in HYSYS.

TABLE 2.2 Properties of the test stream calculated by PVTx simulator.

	Z_i	T_c	P_c	ω	C_a	C_b
Components		°R	psia			
C1	0.642	342.208	666.543	0.012	1.003	1.002
CO2	0.007	547.570	1070.600	0.231	1.000	1.000
C2	0.059	549.760	707.800	0.091	1.000	1.000
C3	0.028	665.680	616.300	0.145	1.000	1.000
C4	0.009	756.588	544.550	0.188	0.985	1.003
C5	0.002	838.549	489.340	0.241	1.009	0.988
C6F3	0.002	1028.432	418.363	0.316	0.966	0.979
F4	0.073	1233.655	310.286	0.480	1.000	1.000
F5F6	0.093	1490.700	263.971	0.710	0.759	0.928
H2O	0.083	1165.140	3208.235	0.344	0.997	1.000

TABLE 2.3 Component properties modified by the proposed regression method.

	Z_i	T'_c	P'_c	ω'
Components		°R	psia	
C1	0.642	342.28043800	665.06915151	0.01177886
CO2	0.007	547.56032893	1070.55242542	0.23098070
C2	0.059	549.75016956	707.77070888	0.09078327
C3	0.028	665.66754115	616.27157865	0.14540422
C4	0.009	748.55238225	537.33885874	0.18392694
C5	0.002	848.65494644	501.09635564	0.24670545
C6F3	0.002	1020.95956776	424.19951859	0.31186892
F4	0.073	1233.58762095	310.26692155	0.47986616
F5F6	0.093	1365.19124228	260.55137287	0.60991481
H2O	0.083	1163.21340687	3202.87709020	0.34314673

As can be seen from **Tables 2.4** and **2.5** the calculated equilibrium constants are almost identical for all three simulators for all components. No equilibrium constant has been reported for the water component by Eclipse simulator since water is considered to be a phase in Eclipse simulator rather than a component. Accordingly, the compositional data from Eclipse is modified in LinkControl by calculating water mole fraction using molar flow rate of the producing water from each well stream. In this way, water is included in total composition and is treated like other components in the

HYSYS Property Package. Obviously, if there is no water production from a well, then the water mole fraction will be zero for that well stream in HYSYS.

Some basic phase properties that can be used to check the validity of calculation in each simulator have been calculated and compared. These calculated phase properties and their deviations have been shown in **Tables 2.6** and **2.7**, respectively. As can be seen from these tables, the density values are nearly the same for all simulators emphasizing the validity of the volume shift parameters used in the simulators. The viscosity values calculated by PVTx and Eclipse are in agreement, while HYSYS viscosity values are still somehow different. This difference can certainly be attributed to the viscosity correlations that have been used in HYSYS simulator and accordingly no better viscosity values can be obtained from this simulator.

TABLES 2.4 Equilibrium constant calculated at 100 bar and 100 °C.

Components	K-Value		
	PVTx	HYSYS	Eclipse
C1	4.31250	4.28813	4.30867
CO2	2.13320	2.12657	2.13181
C2	1.31890	1.31515	1.31826
C3	0.68372	0.68228	0.68355
C4	0.39915	0.39844	0.39903
C5	0.19093	0.19075	0.19087
C6F3	0.04383	0.04385	0.04393
F4	0.00393	0.00398	0.00398
F5F6	0.00032	0.00034	0.00032
H2O	0.20664	0.20773	

PVTx simulator was used to generate the required volume shift parameters for each component for use in HYSYS and Eclipse simulators. HYSYS requires the input of volume shift parameter c in $m^3 / kmole$. The c_i values were calculated using PVTx as $c_i = v_i^{NS} - v_i^S$, where v_i^S and v_i^{NS} are the component molar volumes calculated with and without considering volume shifts, respectively. Because of special implementation of volume shift parameters in HYSYS, the numerical values of volume shift parameters calculated using PVTx should be multiplied by -1 for use in HYSYS

simulator. **Table 2.8** indicates the volume shift parameters for the components participating in the test calculation.

TABLES 2.5 Deviation in equilibrium constant calculation for each component.

Components	Deviation		
	PVTx vs. HYSYS	PVTx vs. Eclipse	HYSYS vs. Eclipse
	%	%	%
C1	0.56843	0.08884	-0.47688
CO2	0.31162	0.06523	-0.24562
C2	0.28511	0.04885	-0.23559
C3	0.21158	0.02449	-0.18669
C4	0.17910	0.03121	-0.14763
C5	0.09515	0.03375	-0.06134
C6F3	-0.05015	-0.23418	-0.18412
F4	-1.27174	-1.28599	-0.01443
F5F6	-6.18123	-0.12923	6.45073
H2O	-0.52538		

TABLE 2.6 Basic properties calculated by each simulator.

		PVTx	HYSYS	Eclipse
Gas Density	kg / m ³	66.26718	66.26311	66.27406
Oil Density	kg / m ³	774.11000	773.28283	773.9853
Gas viscosity	cp	0.01498	0.01590	0.014985
Oil viscosity	cp	1.13800	0.86927	1.136032
Gas Z-factor		0.89960	0.89950	
Gas MW		18.49445	18.49205	
Gas gravity		0.63840	0.63832	

The above mentioned procedure can also be used in simulations where water is not present in the system. The calculation results will improve slightly for such cases. Therefore, it is recommended to prepare another Property Package in HYSYS that does not include water using this method instead of considering a zero water mole fraction in a Property Package that includes water as a component.

TABLE 2.7 Deviation in calculated properties in each simulator.

		PVTx vs. HYSYS	PVTx vs. Eclipse	HYSYS vs. Eclipse
		%	%	%
Gas Density	kg / m ³	0.00613	-0.01039	-0.01652
Oil Density	kg / m ³	0.10697	0.01611	-0.09076
Gas viscosity	cp	-5.81187	-0.03337	6.13506
Oil viscosity	cp	30.91456	0.17323	-23.48198
Gas Z-factor		0.01070		
Gas MW		0.01298		
Gas gravity		0.01298		

Most PVT simulators generate specific water Binary Interaction Parameters (BIP) for both aqueous and non-aqueous phases. In general, depending on the final application of the Property Package, one should consider using either of these BIPs. If the Property Package is to be used in a simulation to model hydrocarbon drying operation, for example, then non-aqueous BIPs should be used since the non-aqueous phase containing traces of water is the important phase. In the test calculation, however, the equilibrium constant values calculated by the simulators show the least deviation upon using zero water BIP values in all simulators.

TABLE 2.8 Volume shifts calculated by PVTx for use in HYSYS.

Components	c	Input in HYSYS
	m ³ / kmole	m ³ / kmole
C1	-0.00427	0.00427
CO2	-0.00219	0.00219
C2	-0.00457	0.00457
C3	-0.00484	0.00484
C4	-0.00521	0.00521
C5	-0.00423	0.00423
C6F3	0.00514	-0.00514
F4	0.01980	-0.01980
F5F6	-0.11670	0.11670
H2O	0.00377	-0.00377

2.7.5 Remarks

No further improvement seems to be possible in Equation of State calculation results obtained from the applications participating in the developed hybrid simulator. However, if the user is aware of the differences that exist in thermodynamic models employed in various simulators participating in such hybrid applications, then the analysis of the results will be carried out in a more educated manner and false interpretation of the results will be avoided.

It should be noted that the consistency in calculation is the central issue here rather than which program produces the most realistic EOS calculation result. Therefore, to obtain reliable simulation results in a full-field study using such hybrid programs, it is recommended to use a PVT simulator to generate the PVT compositional information required by reservoir and process simulators using a similar regression method.

Although tuning techniques like the one explained in previous section will to a great extent solve the problem of calculation inconsistency in full-field studies, the ultimate solution is to employ the same thermodynamic model in all participating applications. The achievement of this goal is complicated because of the fact that each simulator in a hybrid program uses obviously its own thermodynamic model and accordingly it is impossible to have a single thermodynamic model for use in the system. On the other hand, such a thermodynamic model that can simulate all existing phase behaviors from the reservoir to the topside facilities with acceptable accuracy has yet to be developed. Such a thermodynamic model should be able to simulate the behavior of near critical mixtures in the reservoir to cryogenic processes in topside facilities. The existing thermodynamic models have been formulated to model only some of these extreme situations with acceptable accuracy and great effort is required to tune these models to new situations. The introduction of more tuning parameters in some thermodynamic models has made them only more difficult to use. This proves that the solution is probably to adopt new approaches to create more general purpose Equations of State. Therefore, one probably has to wait for development of new generation of thermodynamic models that can accurately model all existing phase behaviors in such complex systems. Until that time, the most feasible solution is to make sure that all participating models carry out the calculations as similar to each other as possible.

Another problem associated with using calculation results from the reservoir simulator in the process simulator is the problem of different characterization that is usually used in reservoir and in process simulation studies. While 8 to 12 components are usually quite enough to simulate the reservoir behavior, 15 to 25 components are usually used in process simulation specially in refining operation. This problem is normally addressed by lumping the original components to obtain a characterization with 8 to 12 components. Such a characterization is usually adequate for reservoir simulation. The reservoir simulation results are then de-lumped for use in process simulator where higher number of components is usually required (see for example Danesh 1990, Faissat 1996, Chornng 1996, and Leibovici 2000). Hoda (2002) reports development of techniques to perform the required translation of results between different applications. His work introduces a very good solution to this problem. Such techniques can be used in hybrid applications like the one presented here to speed up the calculations in reservoir simulator by reducing the number of components and then translate the results into a characterization with higher number of components to achieve the require accuracy in process calculations.

In this study, it was decided to avoid the mentioned problem by keeping the number of components as small as possible to reduce the simulation time in the reservoir unit and as big as possible to have enough accuracy in topside calculations. The characterization used in the test calculation of previous section is an example of typical characterization that has been used in this study.

Chapter 3

Optimization

“Would you tell me, please, which way I ought to go from here?”
“That depends a good deal on where you want to get to,” said the Cat.
“I don’t much care where...” said Alice.
“Then it doesn’t matter which way you go,” said the Cat.
“...so long as I get somewhere,” Alice added as an explanation.
“Oh, you’re sure to do that,” said the Cat, “if you only walk long enough.”

Lewis Carroll: Alice in Wonderland.

Walking long enough to reach somewhere is exactly what we do in most cases when facing an optimization problem. We are sure to find a solution or at least get to somewhere if we search enough of the objective function surface. Optimization strategies however have been developed to add the element of intelligence to the search strategy and reduce the random nature of search for the optimum solution. Among various existing optimization strategies, the genetic algorithm search and optimization technique is introduced in this chapter that ironically begins the search for optimum solution in a completely random manner. The random nature of this search strategy seems to help solving efficiently the optimization problems encountered in petroleum production systems. In such systems, variable discontinuities and system non-linearities generate a very rough optimization surface with numerous local maxima or minima. This increases the risk of convergence to a local rather than the global solution. In this chapter, after a detailed introduction to genetic algorithm search strategies, several optimization case studies relevant to petroleum production systems have been presented. These case studies are intended to develop methodologies for solving full-field optimization problems.

3.1 Introduction and background

A number of important design and operation decisions have to be taken for development of a new discovery. For a typical offshore project, the key decisions can be; capacity of processing unit, number of wells, well tubing sizes, choke settings, well location, production and injection rates, size of exporting pipelines, number of exporting fluid phases, and platform versus sub-sea separation. Field optimization studies will accordingly involve optimization of one or several of such decision variables.

The selection of each or any combination of decision variables has to be made in context of the availability of reliable data and applicability of optimization results. An optimum solution is not necessarily a practical or a stable one.

Most of the key decisions in petroleum industry, like those mentioned above, are of discrete nature and the behavior of the system is nonlinear. These make the optimization problem extremely difficult. In most cases, the complexity of the problem is so extreme that any method that solves it even approximately should be regarded as a success (Hallefjord, 1986).

The petroleum industry has always been an important user of optimization techniques. Regarding the huge capital investment required in petroleum industry, it has always made sense to devote considerable resources into development of optimization models and techniques to facilitate and improve petroleum field development.

The systematic optimization studies in the oil industry initiated shortly after advent of linear programming methods in early fifties. The first application of linear programming in oil industry has been reported by Garvin et al. (1957). From this time to late eighties, the reported optimization studies in petroleum industry have used many other similar techniques such as nonlinear programming and dynamic programming (Foster, 1964) and later on, integer programming (see for example Bohannon, 1970).

From the viewpoint of reservoir description, most of the reported optimization studies can be divided into two major categories. The first category consists of studies in which an analytical expression has been used to model the reservoir unit. In the second category, a reservoir model has been integrated and reservoir performance has been considered in the form of output data.

Because of limited computation capabilities, earlier studies use mostly a simplified analytical reservoir description in their optimization models. Wattenbarger (1970) used a well interface to model the reservoir deliverability. Frair et al. (1975) use a simple material balance to represent the reservoir unit and optimize well location and platform building schedule. Asheim (1980) substituted reservoir simulation equations implicitly into an objective function to reduce the number of independent variables and used this method in several production optimization case studies. Bertrand et al.

(1983) use approximate linearized reservoir models. Instead of employing commonly used superposition, a new linear approximation is determined for each time-step in their model. Nesvold et al. (1996) use the well interface approach of Wattenbarger and couple a linear programming method to reservoir simulation and perform a revenue maximization on Ekofisk field redevelopment plan. Pan et al. (1998) report application of two multivariate interpolation techniques to generate the required reservoir performance to predict the optimal strategies in field development scheduling.

More recent optimization studies mainly belong to the second category and employ more sophisticated reservoir models. Huseby et al. (2000) report development of integrated risk models for reservoir and operational costs. They include the reservoir behavior into the study by taking samples of the reservoir data as a function of time. Sharif et al. (2000) describe an integrated workflow procedure to include complex reservoir behavior using stochastic geological and petrophysical uncertainty modeling. Chow et al. (2000) report development of integrated production models for optimization and risk management. Their research is the only reported study in which a dynamically linked reservoir model has been used, although the details of the linking technique have not been disclosed.

Most of the reported research activities in the area of field optimization suffer from insufficient and inaccurate description of the reservoir component in the developed optimization models. In earlier studies where main focus is on numerical optimization, the reservoir behavior is modeled by an analytical equation that is solved along with other analytical expressions modeling other parts of the system. While performance of most of the system components can be easily modeled with simple analytical relations, the description of reservoir performance with such simple expressions is impossible and the reservoir model outputs that are accordingly obtained are considered questionable. In most of the reported studies there is no mention of the effects of the topside facility parameters on the rest of the system and production data obtained from the simplified reservoir model is directly used in optimization calculation. In this way, the costs and constraints involved in activities of the topside production facility are not included in objective function calculation.

In the optimization studies presented in this chapter, the mentioned shortcomings have been eliminated through application of the developed hybrid tool. Upon utilization of real-time data from the reservoir simulator,

the inclusion of superficial reservoir performance in the optimization model is avoided. The detailed performance of the topside facility can be included in the optimization model in order to carry out more realistic studies. The proposed optimization strategy is proved to have a satisfactory performance even in dealing with complex optimization problems.

3.2 A brief look at available optimization techniques

Depending on the objective function, a problem can be classified as either a local or a global optimization problem. If the objective function is convex on its entire domain then the problem is considered to be a local optimization problem and simple local methods such as simple Newton-based methods are usually able to solve it. If the objective function is non-convex, local methods are unable to converge to the global optimum and global optimization methods should accordingly be used (Gray 1997, and Pardalos 1987). Available global optimization techniques can be broadly classified into deterministic and probabilistic approaches.

Deterministic approaches, also called covering or exact methods, are general-purpose procedures for solving non-convex, constrained, global optimization problems. Depending on the sub-category used, a deterministic procedure normally involves manipulating the analytical expression of the objective function in order to obtain a simplified form or a valid relaxation of the non-convex problem. Branch and Bond, Generalized Benders Decomposition, and Cutting Plane method are main deterministic optimization approaches (Horst 1995, Epperly 1995, Smith 1996, Geoffrion 1972, and Floudas 1995).

Deterministic methods cannot be employed to solve the optimization problems that are presented in this study because of the fact that no analytical expressions have been used here to model any component of the petroleum production system. Instead of analytical equations, the outputs of various simulators are processed in search for the optimum solution.

Probabilistic search is one of the more universal solution methods for complex problems where one cannot determine a priori the sequence of steps leading to solution. Probabilistic or stochastic approaches make use of the objective function values rather than its mathematical characteristics to search for the optimum solution and are therefore particularly useful when analytical expression of the objective function is unavailable or very

complex. Evolutionary algorithms, simulated annealing, and polytope method are the main stochastic approaches (Laarhoven 1987, Aarts 1989, Ingber 1993, Goldberg 1989, and Gen 1997).

Evolutionary algorithms include several subclasses such as genetic algorithm, evolutionary programming, and evolutionary strategies. Among these, the genetic algorithms are perhaps the most widely used in search and optimization studies in various branches of science and engineering (Gen, 1997). Genetic algorithm search strategy has been used throughout this study and is discussed in full details in the coming sections.

Simulated annealing uses the analogy of the thermal equilibrium to search for the global optimum. Annealing is the physical process of heating up a solid until melting occurs and cooling it down to form crystal lattice where free energy is minimal (Aarts, 1989). By establishing a correspondence between the cost function and the free energy, and between the solutions and the physical states and associating a control parameter with the temperature, a global optimization algorithm can be constructed based on simulation of the physical annealing process.

The polytope algorithm is a direct search method that uses function comparisons instead of derivative information. The method starts by evaluating the function value at $n+1$ random points. The function values are sorted and the worst value is substituted with a new value calculated at reflection of the centroid of the rest of the n points. The iteration is stopped when the difference between the worst and the best function value is less than the acceptable tolerance (Horst 1995, and Vitanen 1997).

Sen et al. (1992) compare the performance of genetic algorithms and simulated annealing in generation of stochastic permeability fields. Their work is perhaps the first reported study that employs genetic algorithms in reservoir engineering and is among very few reported studies that compare performance of genetic algorithms and simulated annealing. They introduce a new algorithm for simulated annealing, called heat-bath and compare its performance with the commonly used Metropolis algorithm (Metropolis 1953, Kirkpatrick 1983, and Rothman 1985). They however use a very simple genetic algorithm that involves only crossover and mutation operators. They observe that all three algorithms outperform conventional optimization methods and have comparable performances. They report that the heat-bath method converges faster than Metropolis method for larger

problems. They experience a mixed performance by genetic algorithms. They report that the method can reach very close to the global maximum but requires several trial runs to find the optimum population size. They conclude that computation time of genetic algorithms can be reduced substantially by proper choice of fitness function and auxiliary options. This emphasizes the importance of employing a properly written genetic search routine that includes most of the available genetic operators to increase the convergence rate of genetic algorithms.

In a series of publications, Horne et al. have tested the performance of most of the available optimization techniques in solving various optimization problems of relevance to petroleum engineering. In the first paper of this series, Carroll and Horne (1992) use simple analytical models to describe various components of the petroleum production system and apply Newton-based and polytope methods to maximize the net present value of the system. This study shows inapplicability of gradient search methods in determining the optimum function value even when simplified analytical expressions are used to model the system components. In a related paper, Fujii and Horne (1994) compare the performance of a modified Newton method, polytope method, and genetic algorithms in maximizing the net present value of networked production systems. Although they do not present the detailed results of the genetic algorithm calculations, this study still indicates better performance of the genetic algorithms compared to the other two methods. In a more recent paper, Palke and Horne (1997) study the optimization of production systems that include gas lifting, employing genetic algorithms, polytope, and several modified gradient methods. They again employ simple analytical expressions to model the subsystems but try to modify the search methods. This study once again indicates the robustness of genetic algorithms in solving this class of optimization problems.

It can therefore be concluded that the use of optimization methods that require derivative information about the objective function or the methods that involve manipulation of this function is out of question in this study. Among probabilistic methods, genetic algorithm search strategy is chosen to perform the search for optimum solution in the optimization problems discussed in this study since documented research in this field indicate its acceptable performance in handling multivariate optimization problems (Bittencourt 1997, Fujii 1994, and Palke 1997). Moreover, this study is a research activity and as such can both benefit from and contribute to the ongoing research in the field of genetic algorithms.

This study refrains from comparing different optimization techniques, as the central issue in this research is to demonstrate viability of integrated simulation rather than comparison of existing alternatives.

3.3 Genetic algorithms

Since sixties there has been an increasing interest in simulating the behavior of living species to solve complex optimization problems encountered in various engineering disciplines. A class of stochastic optimization techniques was developed to simulate the natural evolutionary processes in order to search for optimal solution of difficult engineering problems. These techniques are called evolutionary algorithms. As was mentioned earlier, at present the research on this field is conducted on three main branches. These are genetic algorithms, evolutionary programming, and evolutionary strategies. These algorithms usually outperform conventional optimization methods when dealing with difficult optimization problems (Bäck 1996, and Schwefel 1994). At present, genetic algorithms are the most widely known and perhaps the most widely employed algorithm in this class.

Genetic algorithms are robust and widely applicable stochastic search and optimization techniques based on doctrine of evolutionary theory. David E. Goldberg is a pioneer in this field and has documented this algorithm in his classical book on genetic algorithms (Goldberg, 1989). Goldberg defines genetic algorithms in this way: “Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search”.

In recent years, genetic algorithms have found increasing application in various engineering disciplines to solve complex optimization problems. The applications of genetic algorithms in engineering optimization include; scheduling and sequencing, reliability design, vehicle routing, and architectural design.

3.3.1 Structure of simple genetic algorithms (Gen, 1997)

As described originally by Holland (1975) and documented by Goldberg, genetic algorithms start with an initial set of random solutions called *population*. Each individual in the population is called a *chromosome* and represents a solution to the problem. A chromosome is usually a string of binary bits. The chromosomes evolve through successive iterations. Each iteration corresponds to a generation. At each generation, the chromosomes are evaluated and ranked according to their *fitness*. To create the next generation, new chromosomes called *offsprings*, are formed by either merging two chromosomes from current generation using a *crossover* operator or modifying a chromosome using a *mutation* operator. A new generation is formed by selecting some of the parents or offspring chromosomes according to their fitness values and rejecting others so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum or near-optimal solution to the problem.

Grefenstette et al. (1989) describe the sequences involved in a simple genetic algorithm in this way:

Let $P(t)$ and $C(t)$ be parents and offsprings or children in current generation t , the genetic algorithm can then be presented as follows:

```
begin
   $t \leftarrow 0$ 
  initialize  $P(t)$ 
  evaluate  $P(t)$ 
  while (not termination condition) do
    recombine  $P(t)$  to yield  $C(t)$ 
    evaluate  $C(t)$ 
    select  $P(t+1)$  from  $P(t)$  and  $C(t)$ 
     $t \leftarrow t+1$ 
  end
end
```

The above algorithm has been depicted on **Figure 3.1**.

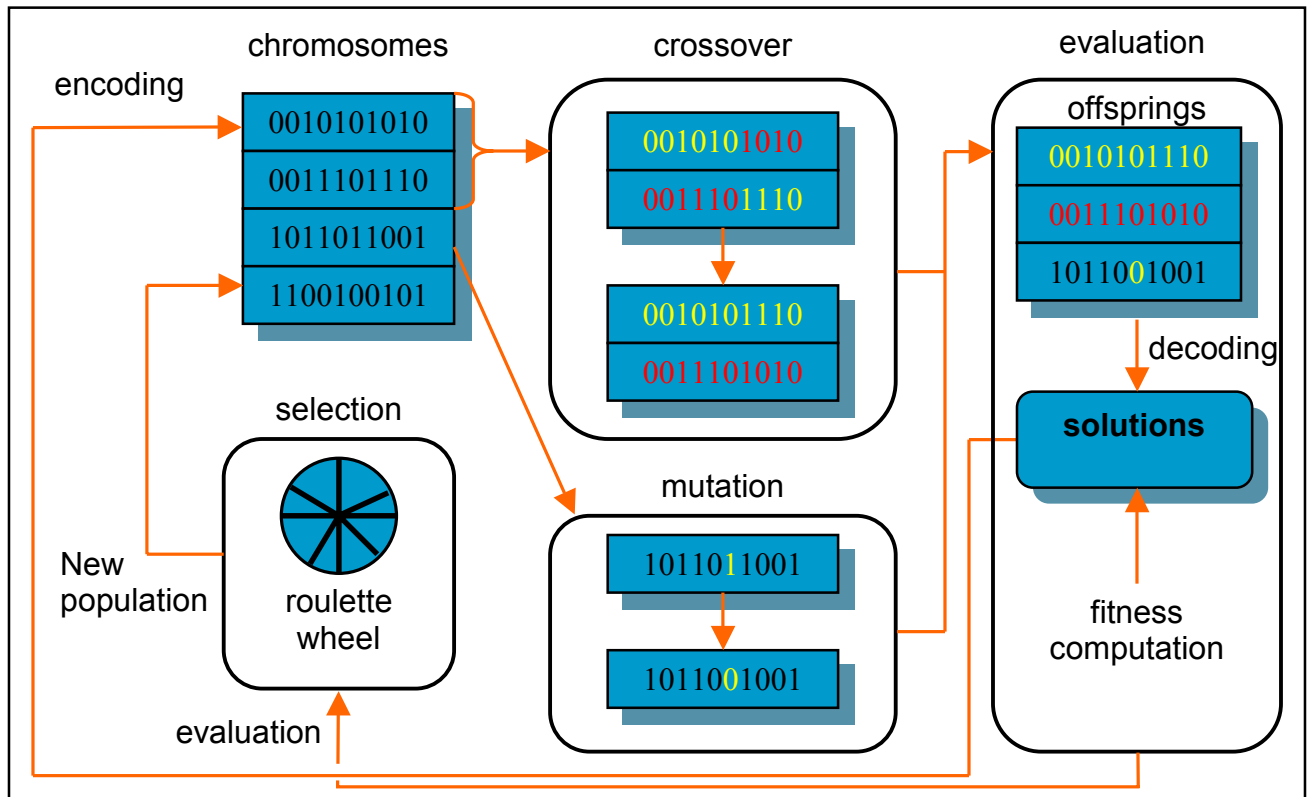


FIGURE 3.1 General structure of genetic algorithms.

The initialization is usually carried out in a random manner. Evaluation is done by assessing the function value calculated for each chromosome. Recombination at least involves crossover and mutation operations to generate offsprings. For more complicated problems, several other genetic operations such as *elitism*, *niching*, and *jump* and *creep* mutations are performed to insure more robust search.

The operations performed on chromosomes can be classified as:

1. Genetic operations, such as crossover and mutation
2. Evolution operation, i.e., selection

The genetic operations simulate natural genetic processes to create new offsprings at each generation. The evolution operation simulates the process of Darwinian evolution to create populations from one generation to another. Crossover is the main genetic operator. It modifies two chromosomes at a time and generates offspring by combining both chromosomes' features. A

simple way to perform crossover would be to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point.

This method is specially suited to the bit string representation of chromosomes. The performance of genetic algorithms depends largely on the performance of the applied crossover operator.

The crossover rate, denoted by p_c is defined as the ratio of the number of offsprings produced in each generation to the population size, denoted by p_s . This ratio controls the expected number $p_c \times p_s$ of chromosomes to undergo the crossover operation. A higher crossover rate allows exploration of more of the solution space and reduces the chances of settling for a local optimum, but if the rate is too high, it results in wasting a lot of computation time in exploring unpromising regions of the solution space.

In artificial genetic systems, the mutation operator protects against irrecoverable loss of valuable genes that are excluded from the population by reproduction and crossover operations. Mutation produces spontaneous random changes in various chromosomes by altering one or more genes. In genetic algorithms, mutation serves the crucial role of either replacing the genes lost from the population during the selection process so that they can be tried in a new context or providing the genes that were not present in the initial population.

The mutation rate denoted by p_m is defined based on the percentage of the total number of genes in the population. The mutation rate controls the rate at which new genes are introduced into the population for trial. If it is too low, many genes that would have been useful are never tried out. On the other hand, if it is too high, there will be so much random genetic modifications that children will start losing their resemblance to the parents and the algorithm will lose the ability to learn from the history of the search.

Genetic algorithm search strategies differ from conventional optimization and search procedures in several fundamental ways. Goldberg summarizes these as follows:

- Genetic algorithms work with a coding of the solution set not the solutions themselves.
- Genetic algorithms search from a population of solutions not a single solution.
- Genetic algorithms use payoff information or fitness function not derivatives or other auxiliary knowledge.
- Genetic algorithms use probabilistic rules not deterministic rules.

3.3.2 Major advantages of genetic algorithms

As was discussed earlier, search is one of the more favorite problem solving techniques for complex problems where a prior knowledge of the sequence of steps leading to solution is not available. In general, search can be performed with either blind strategies or heuristic strategies (Bolc, 1992). Blind search strategies do not use information about the problem domain. Heuristic search strategies use additional information to guide the search along with the best search directions. Exploiting the best solution and exploring the search space are the two important issues in search strategies. Michalewicz (1994) gives a comparison of hill-climbing search, random search and genetic search. Hill-climbing is an example of a strategy which exploits the best direction information for possible improvement while fails to explore the search space. Random search on the other hand explores the search space while ignores the exploitation of the promising regions of the search space. Genetic algorithms are a class of general-purpose search methods that combine advantages of directed and stochastic search that make a notable balance between exploitation and exploration of the search space. Genetic search begins with a randomly generated population. Crossover operator begins to perform widespread search for exploring all solution space. As the high fitness solutions develop, the crossover operator provides exploration in the neighborhood of each of them. The type of the search a crossover operator performs is determined by the environment of the genetic system not by the operator itself. Simple genetic operators are basically designed as general purpose or domain independent search methods. They perform essentially a blind search and could not guarantee to yield improved offsprings.

Most classical optimization methods generate a deterministic sequence of computation based on the gradient or higher order derivatives of objective function. These methods are applied to a single point in the search space. The point is then improved along the highest derivative gradually through

iterations. This point-to-point approach takes the danger of falling in local optima. Genetic algorithms perform a multidirectional search by maintaining a population of potential solutions. The population-to-population approach that includes testing a number of solutions simultaneously, attempts to make the search escape from local optima. Population undergoes a simulated evolution. At each generation the relatively good solutions are reproduced, while the relatively bad solutions die. Genetic algorithms use probabilistic rules to select someone to be reproduced and someone to die so as to guide their search towards regions of the search space with higher likelihood of finding the optimum solution.

Genetic algorithms have received considerable attention regarding their potential as a novel optimization technique. There are three major advantages when applying genetic algorithms to optimization problems (Gen, 1997):

1. Genetic algorithms do not have much mathematical requirements about the optimization problems. Due to their evolutionary nature, genetic algorithms will search for solutions without regard to the specific inner workings of the problem. Genetic algorithms can handle any kind of objective functions and any kind of constraints, i.e., linear or nonlinear, defined on discrete, continuous or mixed search spaces.
2. The application of evolution operators makes genetic algorithms very effective at performing global search in probability. The traditional approaches perform local search by a convergence stepwise procedure, which compares the values of nearby points and moves to the relative optimal points. Global optima can be found only if the problem possesses certain convexity properties that essentially guarantee that any local optimum is the global optimum. As depicted on **Figure 3.2**, while hill-climbing search from point *A* will probably converge to the global optimum, a search from point *B* will certainly be trapped in the local optimum. Genetic search begins at several random points and therefore is less likely to be trapped in a local optimum.
3. Genetic algorithms provide a great flexibility to make an efficient implementation for a specific search surface.

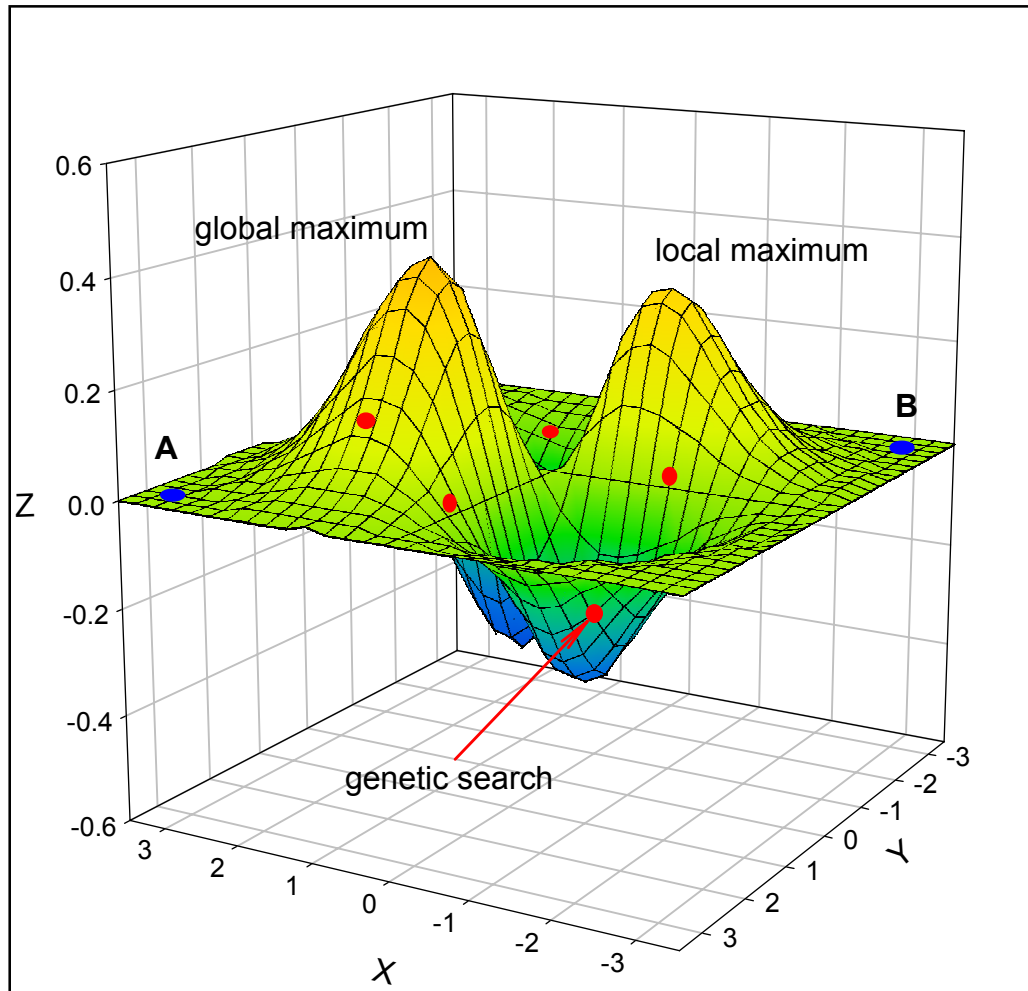


FIGURE 3.2 Random nature of search avoids convergence to local optima in genetic search.

3.3.3 Vocabulary

Genetic algorithms are rooted in both natural genetics and computer science, the terminologies used in genetic algorithm literature are, therefore, a mixture of the natural and artificial.

In a biological organism, the structure that encodes the prescription specifying how the organism is to be constructed is called a *chromosome*. One or more chromosomes may be required to specify the complete organism. The complete set of chromosomes is called a *genotype*, and the resulting organism is called a *phenotype*. Each chromosome comprises a number of individual structures called *genes*. Each gene encodes a particular

feature of the organism, and the location or locus of the gene within the chromosome structure determines what particular characteristic the gene represents. At a particular locus, a gene may encode any of several different values of the particular characteristic it represents. The different values of a gene are called *alleles*. The correspondence of genetic algorithm terms and optimization terms is summarized in **Table 3.1**.

TABLE 3.1 Explanation of genetic algorithm terms.

Genetic Algorithm Terms	Explanation
Chromosome, String, Individual	Solution, Coding
Genes, Bits	Parts of solution
Locus	Position of gene
Alleles	Values of gene
Phenotype	Decoded solution
Genotype	Encoded solution

3.3.4 Special operators in genetic algorithms

In the discussions presented so far only three basic genetic operators were explained. However, to address more complicated optimization problems, the use of some natural operators and phenomena are inevitable. In this section, some advanced genetic algorithm operators that have been employed in this study will be discussed. This is not meant to be a comprehensive explanation of the topic but rather a background preparation for coming discussions. In a chapter devoted to this topic, Goldberg (1989) elaborates on philosophy and applications of these advanced operators.

3.3.4.1 Niche and speciation

In natural context, niching may be viewed as an organism's job or role in the environment. Species are viewed as a class of organism with common characteristics. In the context of genetic algorithm, niches are defined as sub-domains of a function and species are considered as stable subpopulation of strings. The implementation of niche and species can help genetic algorithm search to converge to more promising regions of the solution space by taking into consideration the relative fitness calculated for each function sub-domain.

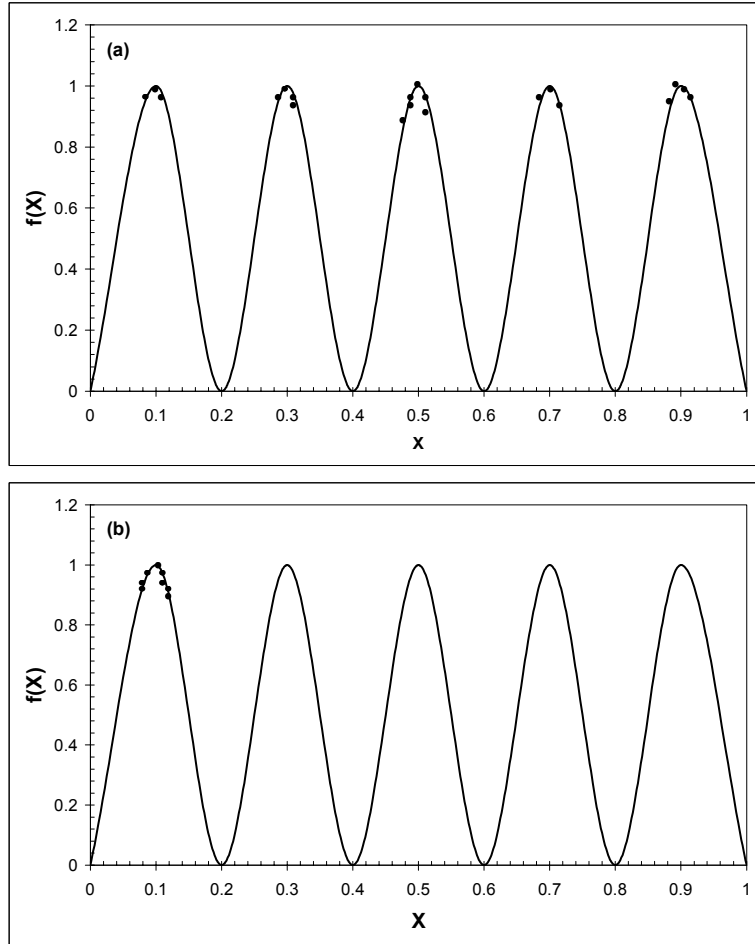


FIGURE 3.3 Performance of simple genetic algorithm on equal peaks, (a) with and (b) without sharing.

To clarify this matter, the action of a simple genetic algorithm on the simple function depicted in **Figure 3.3** is considered. Starting from a random initial population, a relatively even spread of points across the function domain will be obtained. As reproduction, crossover and mutation proceed, the population climbs the hills, ultimately distributing most of the strings near the top of one hill among the five. This ultimate convergence on one peak or another without differential advantage is caused by genetic drift or stochastic sampling caused by small population sizes. Evidently, the preferred convergence scheme in such cases is formation of stable subpopulations around each equal peak.

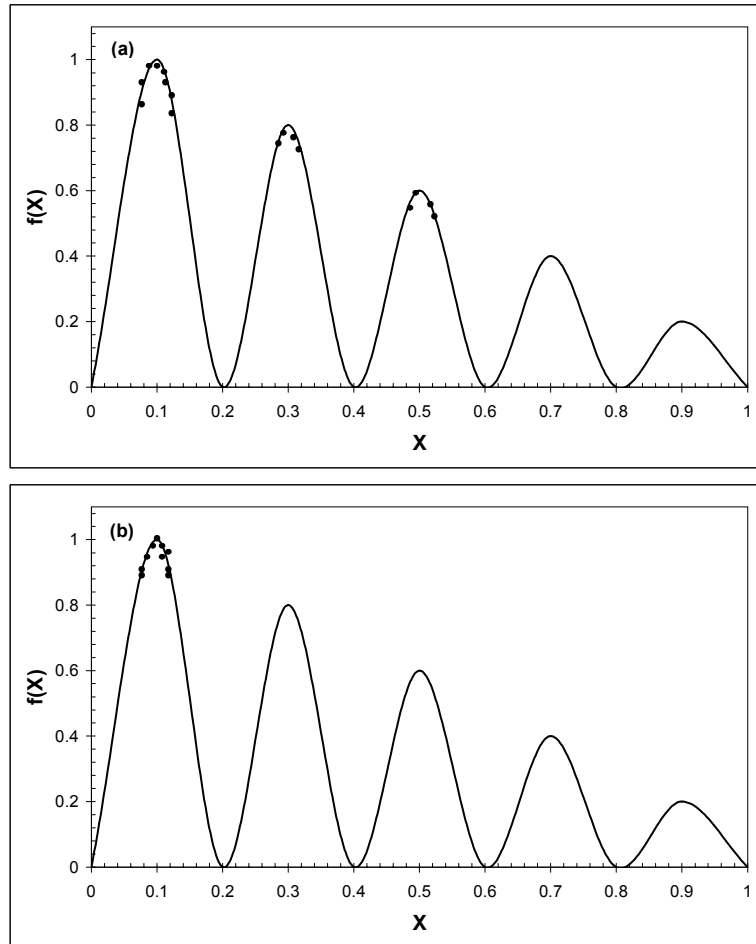


FIGURE 3.4 Performance of simple genetic algorithm on decreasing peaks, (a) with and (b) without sharing.

Sometimes the performance of simple genetic algorithm is required to be modified in order to decrease the subpopulation sizes with decreasing calculated fitness. For example, the performance of simple genetic algorithm in solving the function presented in **Figure 3.4** is to distribute almost all of its points about the highest peak. In general, allocating subpopulations to peaks in proportion to peak magnitude is the preferred scheme. In this way, more of the solution space is searched while fast convergence to the global optimum is guaranteed.

The inducement of niche and species can help genetic algorithm to conduct the search in the manner discussed above. In most reported researches carried out on genetic algorithm, niche-like behavior have been implemented by forbidding mating between dissimilar species.

Replacement of parents by superior offsprings in order to avoid population size increase and maintain the diversity in population simulates a niche-like behavior as well. Goldberg et al. (1987) present a practical scheme that uses a sharing metaphor first discussed by Holland (1975). In this scheme, a sharing function is defined to determine the neighborhood and degree of sharing for each string in the population.

For simple functions, like those presented in **Figures 3.3** and **3.4**, and a simple sharing function depicted in **Figure 3.5**, the method works by determining the degree of sharing for an individual by summing the sharing function values contributed by all other strings in the population. Strings close to an individual require a high degree of sharing (close to one), and strings far from the individual require a low degree of sharing (close to zero). Since an individual is very close to itself, its sharing function value is one. After accumulating the total number of shares in this manner, an individual's derated fitness is calculated by taking the potential fitness, i.e., the unshared value, and dividing through by the accumulated number of shares as follows:

$$f_s(x_i) = \frac{f(x_i)}{\sum_{j=1}^n s[d(x_i, x_j)]} \quad 3.1$$

Therefore, when many individuals are in the same neighborhood they contribute to one another's share count, thereby derating one another's fitness values. As a result, this mechanism limits the uncontrolled growth of particular species within a population.

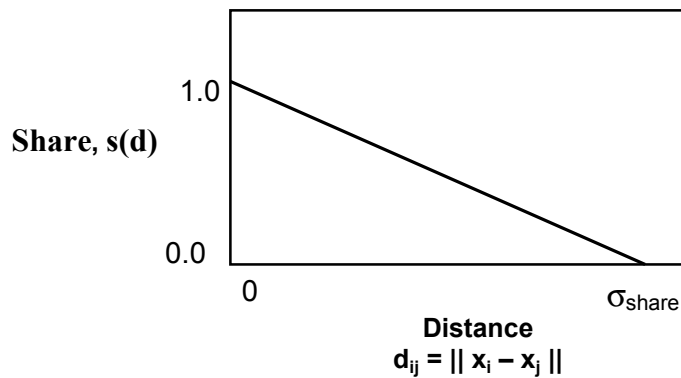


FIGURE 3.5 Triangular sharing function, after Goldberg and Richardson description (1987).

3.3.4.2 Elitism

This operator was first suggested by De Jong (1975). It tends to preserve the individuals with highest fitness value or distinct superior properties for inclusion in new generation. De Jong's expression of elitism is as follows:

Let $a^*(t)$ be the best individual up to the time t . If, after generating the population $A(t+1)$ in the usual fashion, $a^*(t)$ is not in $A(t+1)$, then include $a^*(t)$ to $A(t+1)$ as the $(n+1)$ th member.

De Jong points out that elitism improves local search at the expense of global perspective.

3.3.4.3 Shuffling and inversion operator

Inversion operator changes the ordering of genes on chromosomes by selecting two points along the length of chromosome, cutting chromosome at those points and switching places of the cut section. For example, consider the following chromosome where two inversion sites are chosen at random as shown:

10[^]11110[^]11

The use of a simple inversion operator would obtain the following string:

10011111

In simple genetic algorithm formulations, fitness value is not a function of gene location, therefore, inversion has no direct effect on string fitness. However, inversion is useful in searching for good string arrangements at the same time other genetic operators are searching for good allele sets. If the current population contains bad ordering, crossover will destroy important allele packets with high probability.

3.3.4.4 Micro operators

A number of low-level operators have been suggested for use in genetic adaptive search. Although these add marginal power to genetic algorithms as compared to other advanced operators, their use is common in genetic

algorithm routines. Segregation and Translocation are two such operators that have been implemented in this study and are explained briefly here.

Segregation operator disrupts any linkage that might exist between genes on different chromosomes by randomly selecting and modifying a micro population. Segregation is a useful operator if relatively independent genes happen to have located themselves on different chromosomes. In this way, with linkage effectively destroyed, poor alleles cannot take undue credit of the strength of highly fit, unrelated alleles and take way to survival again.

Translocation can be viewed as an inter-chromosomal crossover operator that organizes the genes in chromosome in an appropriate manner. To implement translocation operator, alleles with their gene name should be tagged so that their intended meaning can be identified when they are shuffled from chromosome to chromosome by the translocation operator.

3.3.5 Numerical example

The actual working principle of genetic algorithms becomes much clearer by looking at a numerical example. In this section, major steps involved in a simple genetic algorithm are demonstrated using a numerical example.

3.3.5.1 Optimization problem

The unconstrained optimization test problem is defined in this way:

$$\text{maximize } f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2) \quad 3.2$$

$$\text{where } \begin{aligned} -3.0 \leq x_1 \leq 12.1 \\ 4.1 \leq x_2 \leq 5.8 \end{aligned} \quad 3.3$$

This is a classic example and has been used by several authors to explain the inner working of simple genetic algorithms (see for example Michalewicz 1994, and Gen 1997). Part of the objective function surface is shown in **Figure 3.6**.

3.3.5.2 Representation

The first step in implementing the genetic algorithm is to encode decision variables into binary strings. The length of the string depends on the required precision. The domain of variable x_j is assumed to be $[a_j, b_j]$ and the required precision is four decimal places for each variable. The precision requirement implies that the range of domain of each variable should be divided into at least $(b_j - a_j) \times 10^4$ size ranges. The required bits, denoted by m_j , for a variable is calculated as follows:

$$2^{m_j-1} < (b_j - a_j) \times 10^4 \leq 2^{m_j} \quad 3.4$$

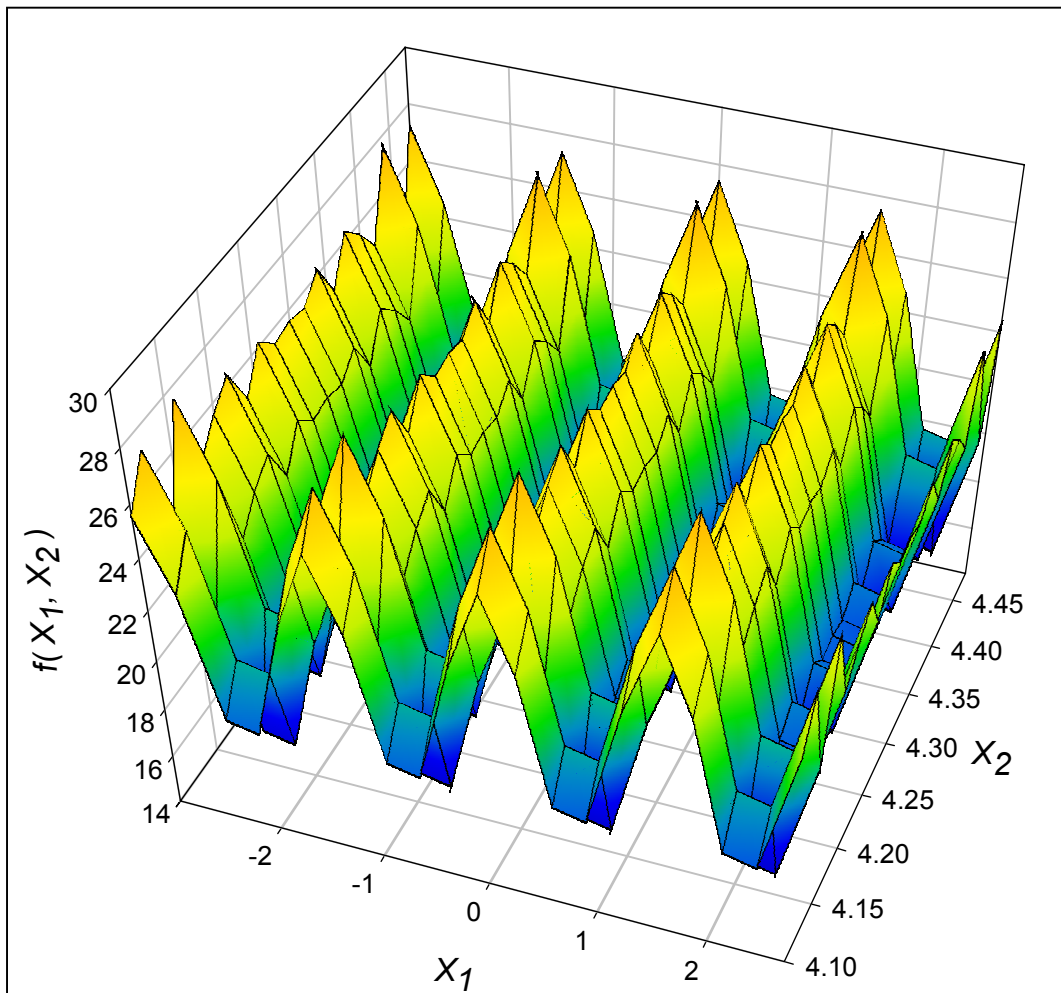


FIGURE 3.6 Rough surface of optimization function presented in the demonstration example.

The mapping from a binary string to a real number for variable x_j is straightforward as follows:

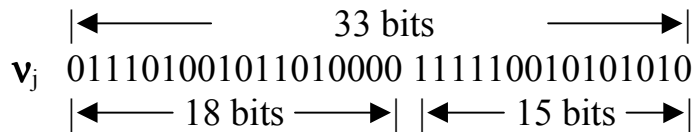
$$x_j = a_j + \text{decimal}(\text{substring}_j) \times \frac{b_j - a_j}{2^{m_j} - 1} \quad 3.5$$

where $\text{decimal}(\text{substring}_j)$ represents the decimal value of substring_j for decision variable x_j .

If the precision is set at four decimal places, the required bits for variables x_1 and x_2 are calculated in this way:

$$\begin{aligned} (12.1 - (-3.0)) \times 10,000 &= 151,000 & 3.6 \\ 2^{17} < 151,000 \leq 2^{18} &\Rightarrow m_1 = 18 \\ (5.8 - 4.1) \times 10,000 &= 17,000 \\ 2^{14} < 17,000 \leq 2^{15} &\Rightarrow m_2 = 15 \\ m = m_1 + m_2 &= 18 + 15 = 33 \end{aligned}$$

The total length of a chromosome is, therefore, 33 bits that can be represented by this randomly generated sample:



The corresponding values of example variables x_1 and x_2 are then given as:

	Binary number	Decimal number
x_1	011101001011010000	119504
x_2	111110010101010	31914

therefore,

$$x_1 = -3.0 + 119504 \times \frac{12.1 - (-3.0)}{2^{18} - 1} = 3.883687 \quad 3.7$$

$$x_2 = 4.1 + 31914 \times \frac{5.8 - 4.1}{2^{15} - 1} = 5.755745$$

3.3.5.3 Initial population

Initial population is generated randomly as follows. The population size is assumed to be 10.

$$\begin{aligned} \mathbf{v}_1 &= [001001010100101001101111011111110] & 3.8 \\ \mathbf{v}_2 &= [111000100100110111001010100011010] \\ \mathbf{v}_3 &= [000010000011001000001010111011101] \\ \mathbf{v}_4 &= [100110110100101101000000010111001] \\ \mathbf{v}_5 &= [000010111101100010001110001101000] \\ \mathbf{v}_6 &= [111110101011011000000010110011001] \\ \mathbf{v}_7 &= [110100010011111000100110011101101] \\ \mathbf{v}_8 &= [100001100001110100010110101100111] \\ \mathbf{v}_9 &= [111110001011101100011101000111101] \\ \mathbf{v}_{10} &= [000001111000110000011010000111011] \end{aligned}$$

These correspond to the following decimal values:

$$\begin{aligned} \mathbf{v}_1 &= [x_1, x_2] = [-0.800462, 5.361653] & 3.9 \\ \mathbf{v}_2 &= [x_1, x_2] = [10.348434, 4.380264] \\ \mathbf{v}_3 &= [x_1, x_2] = [-2.516603, 4.390381] \\ \mathbf{v}_4 &= [x_1, x_2] = [6.159951, 4.109598] \\ \mathbf{v}_5 &= [x_1, x_2] = [-2.301286, 4.477282] \\ \mathbf{v}_6 &= [x_1, x_2] = [11.788084, 4.174346] \\ \mathbf{v}_7 &= [x_1, x_2] = [9.342067, 5.121702] \\ \mathbf{v}_8 &= [x_1, x_2] = [4.910618, 4.703018] \\ \mathbf{v}_9 &= [x_1, x_2] = [11.671267, 4.873501] \\ \mathbf{v}_{10} &= [x_1, x_2] = [-2.554851, 4.793707] \end{aligned}$$

Procedure: Evaluation

The fitness of a chromosome is evaluated through the following three steps:

Step 1. Convert the chromosome's genotype to its phenotype. This means converting binary string into relative real values $x^k = (x_1^k, x_2^k)$, $k = 1, 2, \dots, p_s$.

Step 2. Evaluate the object function $f(x^k)$.

Step 3. Convert the value of objective function into fitness. Here the fitness is simply equal to the value of objective function:

$$eval(v_k) = f(x^k), \quad k = 1, 2, \dots, p_s.$$

The evaluation function ranks the chromosomes according to their fitness values.

The fitness values of above chromosomes are as follows:

$$\begin{aligned} eval(\mathbf{v}_1) &= f(-0.800462, 5.361653) &= 19.396331 & 3.10 \\ eval(\mathbf{v}_2) &= f(10.348434, 4.380264) &= 7.580015 \\ eval(\mathbf{v}_3) &= f(-2.516603, 4.390381) &= 19.526329 \\ eval(\mathbf{v}_4) &= f(6.159951, 4.109598) &= 29.406122 \\ eval(\mathbf{v}_5) &= f(-2.301286, 4.477282) &= 15.686091 \\ eval(\mathbf{v}_6) &= f(11.788084, 4.174346) &= 11.900541 \\ eval(\mathbf{v}_7) &= f(9.342067, 5.121702) &= 17.958717 \\ eval(\mathbf{v}_8) &= f(4.910618, 4.703018) &= 17.959701 \\ eval(\mathbf{v}_9) &= f(11.671267, 4.873501) &= 26.401669 \\ eval(\mathbf{v}_{10}) &= f(-2.554851, 4.793707) &= 21.278435 \end{aligned}$$

Evidently, chromosome \mathbf{v}_4 is the strongest and chromosome \mathbf{v}_2 the weakest.

3.3.5.4 Selection

In most formulations, a roulette wheel approach is adopted as the selection procedure. It selects chromosomes based on proportion of fitness and can select new population with respect to the probability distribution based on fitness values. The roulette wheel can be constructed as follows:

1. Calculate the fitness value $eval(\mathbf{v}_k)$ for each chromosome \mathbf{v}_k

$$eval(v_k) = f(x), \quad k = 1, 2, \dots, p_s \quad 3.11$$

2. Calculate the total fitness for the population

$$F = \sum_{k=1}^{p_s} eval(\mathbf{v}_k) \quad 3.12$$

3. Calculate selection probability p_k for each chromosome \mathbf{v}_k

$$p_k = \frac{eval(\mathbf{v}_k)}{F}, \quad k = 1, 2, \dots, p_s \quad 3.13$$

4. Calculate cumulative probability q_k for each chromosome \mathbf{v}_k

$$q_k = \sum_{j=1}^k p_j, \quad k = 1, 2, \dots, p_s \quad 3.14$$

The selection process begins by spinning the roulette wheel p_s times, each time a single chromosome is selected for a new population in the following way:

Procedure: Selection

Step 1. Generate a random number r from the range $[0, 1]$.

Step 2. If $r \leq q_1$, then select the first chromosome \mathbf{v}_1 , otherwise, select the k th chromosome \mathbf{v}_k ($2 \leq k \leq p_s$) such that $q_{k-1} \leq r \leq q_k$.

The total fitness F of the population is:

$$F = \sum_{k=1}^{10} eval(\mathbf{v}_k) = 187.093951 \quad 3.15$$

The probability of a selection p_k for each chromosome \mathbf{v}_k ($k = 1, \dots, 10$) is as follows:

$$\begin{array}{lll} p_1 = 0.103672 & p_2 = 0.040514 & p_3 = 0.104366 \\ p_4 = 0.157173 & p_5 = 0.083841 & p_6 = 0.063607 \\ p_7 = 0.095988 & p_8 = 0.095993 & p_9 = 0.141114 \\ p_{10} = 0.113731 & & \end{array}$$

The cumulative probability q_k for each chromosome \mathbf{v}_k ($k = 1, \dots, 10$) is as follows:

$$\begin{array}{lll}
q_1 = 0.103672 & q_2 = 0.144186 & q_3 = 0.248552 \\
q_4 = 0.405725 & q_5 = 0.489566 & q_6 = 0.553173 \\
q_7 = 0.649161 & q_8 = 0.745154 & q_9 = 0.886268 \\
q_{10} = 1.00000 & &
\end{array}$$

Now the roulette wheel is turned 10 times and each time a single chromosome is selected for a new population. A random sequence of 10 numbers from the range [0, 1] can be generated for example as:

$$\begin{array}{llll}
r_1 = 0.301431 & r_2 = 0.322062 & r_3 = 0.766503 & r_4 = 0.881893 \\
r_5 = 0.350871 & r_6 = 0.583392 & r_7 = 0.177618 & r_8 = 0.343242 \\
r_9 = 0.032685 & r_{10} = 0.197577 & &
\end{array}$$

The first number $r_1 = 0.301431$ is greater than q_3 and smaller than q_4 , meaning that the chromosome \mathbf{v}_4 is selected for the new population. The second number $r_2 = 0.322062$ is greater than q_3 and smaller than q_4 , meaning that the chromosome \mathbf{v}_4 is again selected for the new population, and so on. Finally, the new population consists of the following chromosomes

$$\begin{array}{ll}
\mathbf{v}'_1 = [100110110100101101000000010111001] & (\mathbf{v}_4) \\
\mathbf{v}'_2 = [100110110100101101000000010111001] & (\mathbf{v}_4) \\
\mathbf{v}'_3 = [111110001011101100011101000111101] & (\mathbf{v}_9) \\
\mathbf{v}'_4 = [111110001011101100011101000111101] & (\mathbf{v}_9) \\
\mathbf{v}'_5 = [100110110100101101000000010111001] & (\mathbf{v}_4) \\
\mathbf{v}'_6 = [110100010011111000100110011101101] & (\mathbf{v}_7) \\
\mathbf{v}'_7 = [000010000011001000001010111011101] & (\mathbf{v}_3) \\
\mathbf{v}'_8 = [100110110100101101000000010111001] & (\mathbf{v}_4) \\
\mathbf{v}'_9 = [00100101010010100110111101111110] & (\mathbf{v}_1) \\
\mathbf{v}'_{10} = [000010000011001000001010111011101] & (\mathbf{v}_3)
\end{array} \quad 3.16$$

3.3.5.5 Crossover

Crossover used here is one-cut-point method, which randomly selects a cut-point and exchanges the right parts of two parents to generate offspring. As an example, crossover operator works in this way on the 17th gene of two typical chromosomes:

$$\begin{array}{c}
\downarrow \\
\mathbf{v}'_1 = [100110110100101101000000010111001] \\
\mathbf{v}'_3 = [111110001011101100011101000111101]
\end{array}$$

The resulting offsprings by exchanging the right parts of their parents would be as follows:

$$\mathbf{v}'_1 = [100110110100101100011101000111101]$$

$$\mathbf{v}'_3 = [11111000101110110100000010111001]$$

The probability of crossover is set at $p_c = 0.25$, so that on average 25% of chromosomes undergo crossover. Crossover is performed in the following way:

Procedure: Crossover

begin

$k \leftarrow 0$

while ($k \leq 10$) **do**

$r_k \leftarrow$ random number from $[0, 1]$

if ($r_k < 0.25$) **then**

 select \mathbf{v}_k as one parent for crossover

end

$k \leftarrow k+1$

end

end

If the sequence of random numbers is like this,

$$\begin{array}{llll} r_1 = 0.625721 & r_2 = 0.266823 & r_3 = 0.288644 & r_4 = 0.295114 \\ r_5 = 0.163274 & r_6 = 0.567461 & r_7 = 0.085940 & r_8 = 0.392865 \\ r_9 = 0.770714 & r_{10} = 0.548656 & & \end{array}$$

then this means that the chromosomes \mathbf{v}'_5 and \mathbf{v}'_7 were selected for crossover, since these two numbers are less than 0.25. A random integer number pos is generated from the range $[1, 32]$ as cutting point. This range is selected since 33 is the total length of a chromosome. If pos is for example 1, the two chromosomes are cut after the first bit and offsprings are generated as follows:

$$\mathbf{v}'_5 = [1001101101001011101000000010111001]$$

$$\mathbf{v}'_7 = [000010000011001000001010111011101]$$



$$\mathbf{v}'_5 = [100010000011001000001010111011101]$$

$$\mathbf{v}'_7 = [0001101101001011101000000010111001]$$

3.3.5.6 Mutation

Mutation alters one or more genes with a probability equal to the mutation rate. If the 18th gene of the chromosome \mathbf{v}'_1 is selected for a mutation, the following new chromosome is generated:

$$\mathbf{v}'_1 = [1001101101001011101000000010111001]$$



$$\mathbf{v}'_1 = [10011011010010110000000010111001]$$

The probability of mutation is set at $p_m = 0.01$, so that on average 1% of total bit of population would undergo mutation. There are $m \times p_s = 33 \times 10 = 330$ bits in the whole population. This means that 3.3 mutations per generation. Every bit has an equal chance to be mutated. Thus a sequence of random numbers r_k ($k = 1, \dots, 330$) in the range $[0, 1]$ should be generated. If $r_k < 0.01$ the bit will be mutated.

If as a result of this mutation operator, the following genes undergo mutation,

bit position	v	bit	r_k
105	4	6	0.009857
164	5	32	0.003113
199	7	1	0.000946
329	10	32	0.001282

then the following population is obtained after mutation:

$$\begin{aligned}
\mathbf{v}'_1 &= [100110110100101101000000010111001] & 3.17 \\
\mathbf{v}'_2 &= [100110110100101101000000010111001] \\
\mathbf{v}'_3 &= [111110001011101100011101000111101] \\
\mathbf{v}'_4 &= [111111001011101100011101000111101] \\
\mathbf{v}'_5 &= [100010000011001000001010111011111] \\
\mathbf{v}'_6 &= [110100010011111000100110011101101] \\
\mathbf{v}'_7 &= [100110110100101101000000010111001] \\
\mathbf{v}'_8 &= [100110110100101101000000010111001] \\
\mathbf{v}'_9 &= [001001010100101001101111011111110] \\
\mathbf{v}'_{10} &= [000010000011001000001010111011111]
\end{aligned}$$

The corresponding decimal values of variables $[x_1, x_2]$ and fitnesses are as follows:

$$\begin{aligned}
f(6.159951, 4.109598) &= 29.406122 & 3.18 \\
f(6.159951, 4.109598) &= 29.406122 \\
f(11.671267, 4.873501) &= 23.981506 \\
f(11.907206, 4.873501) &= 5.702781 \\
f(5.033426, 4.390484) &= 21.627295 \\
f(9.342067, 5.121702) &= 17.958717 \\
f(6.159951, 4.109598) &= 29.406122 \\
f(6.159951, 4.109598) &= 29.406122 \\
f(-0.800462, 5.361653) &= 19.534735 \\
f(-2.516603, 4.390484) &= 18.452060
\end{aligned}$$

This constitutes the first iteration in this simple genetic algorithm. After iteration number 419, there is no further increase in calculated fitnesses and the iteration is stopped. The best chromosome at generation 419 is:

$$\begin{aligned}
\mathbf{v}^* &= [111110000000111000111101001010110] & 3.19 \\
eval(\mathbf{v}^*) &= f(11.631407, 5.724824) = 38.81208 \\
x^*_1 &= 11.631407 \\
x^*_2 &= 5.724824 \\
f(x^*_1, x^*_2) &= 38.81208
\end{aligned}$$

Therefore, 38.81208 is the global maximum of the optimization function at point (11.631407, 5.724824).

3.4 VisualGene

VisualGene is a genetic algorithm routine developed in Visual Basic for the purpose of this study based on a Fortran code called GA developed by David L. Carroll (Genetic Algorithm (GA) documentation). VisualGene includes all original features of GA such as elitism, micro-GA, niching, uniform crossover, and creep and jump mutations. Some new options not existing in GA have been included in VisualGene such as possibility of inter-process communication using Automation technique and performing constrained optimization by inclusion of penalty function. The development of VisualGene in Visual Basic made it possible to integrate this optimization code with the rest of the system using Automation. Similar to LinkControl, VisualGene is able to communicate directly with HYSYS and use HYSYS calculation results in optimization studies. It can also control the execution of LinkControl and make the whole hybrid application to perform the iterations required in optimization studies that involve the reservoir unit.

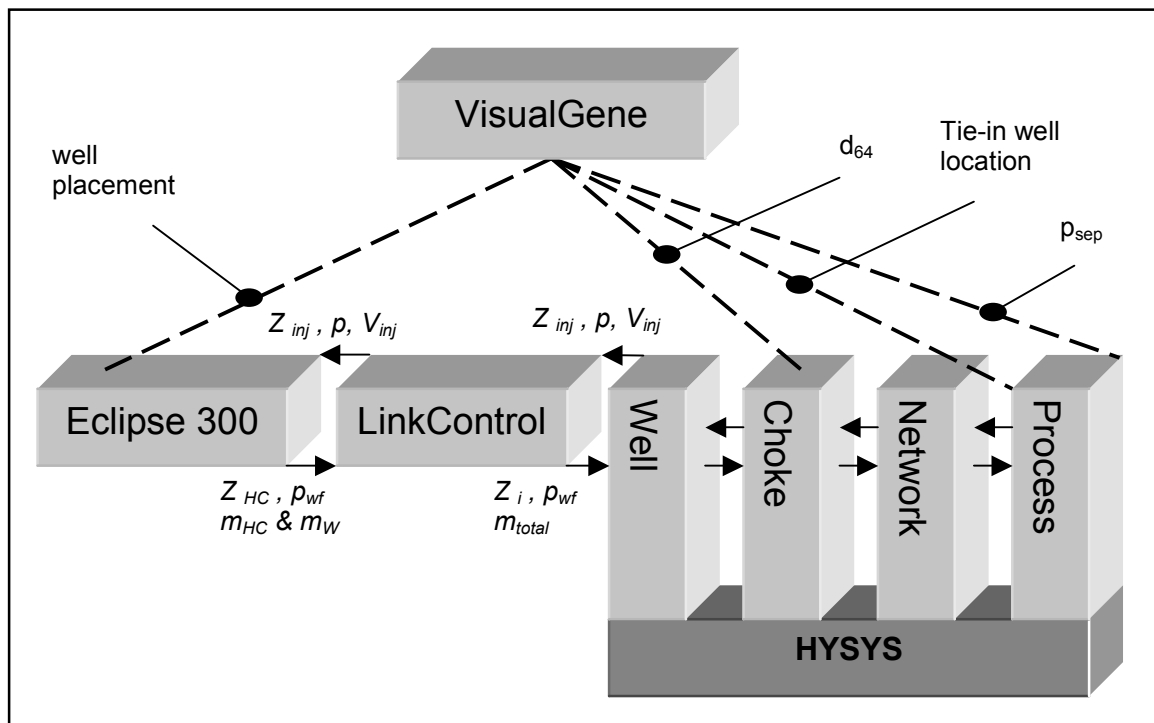


FIGURE 3.7 Integration of VisualGene with the field simulator.

Figure 3.7 depicts the integration of VisualGene with rest of the system. A number of decision variables can be selected from the production system for optimization purposes. These can be reservoir parameters such as well placement, parameters related to topside facility such as pressures at first and second separation stages, and optimal sizing of the surface chokes. It is also possible to programmatically optimize the topside process configuration according to operation requirements. An example is optimum allocation of tie-in satellite wells.

Some of the above mentioned optimization studies have been carried out in the coming sections using VisualGene working either independently or as an integrated part of the hybrid application. Each problem requires some minor modifications to the original VisualGene algorithm, but overall structure of the program is unchanged.

3.4.1 A test problem for VisualGene

Before using VisualGene in actual optimization studies, it is necessary to demonstrate the validity of VisualGene calculations in a relatively difficult test problem.

3.4.1.1 The test function

In this section, the performance of VisualGene to solve a relatively difficult unconstrained optimization problem is demonstrated. The test function is a modified version of the Ackley's function depicted in **Figure 3.8** (Ackley, 1987).

This test function is obtained by modulating an exponential function with a cosine wave of moderate amplitude. Its topology is characterized by a rough outer region and a central peak where modulations by the cosine wave become more and more influential.

This function causes complications to the search through a strictly local hill-climbing optimization algorithm. Such algorithm is surely trapped in a local optimum. A genetic search strategy scans a bigger neighborhood and is able to cross intervening valleys of the test function to find the global optimum.

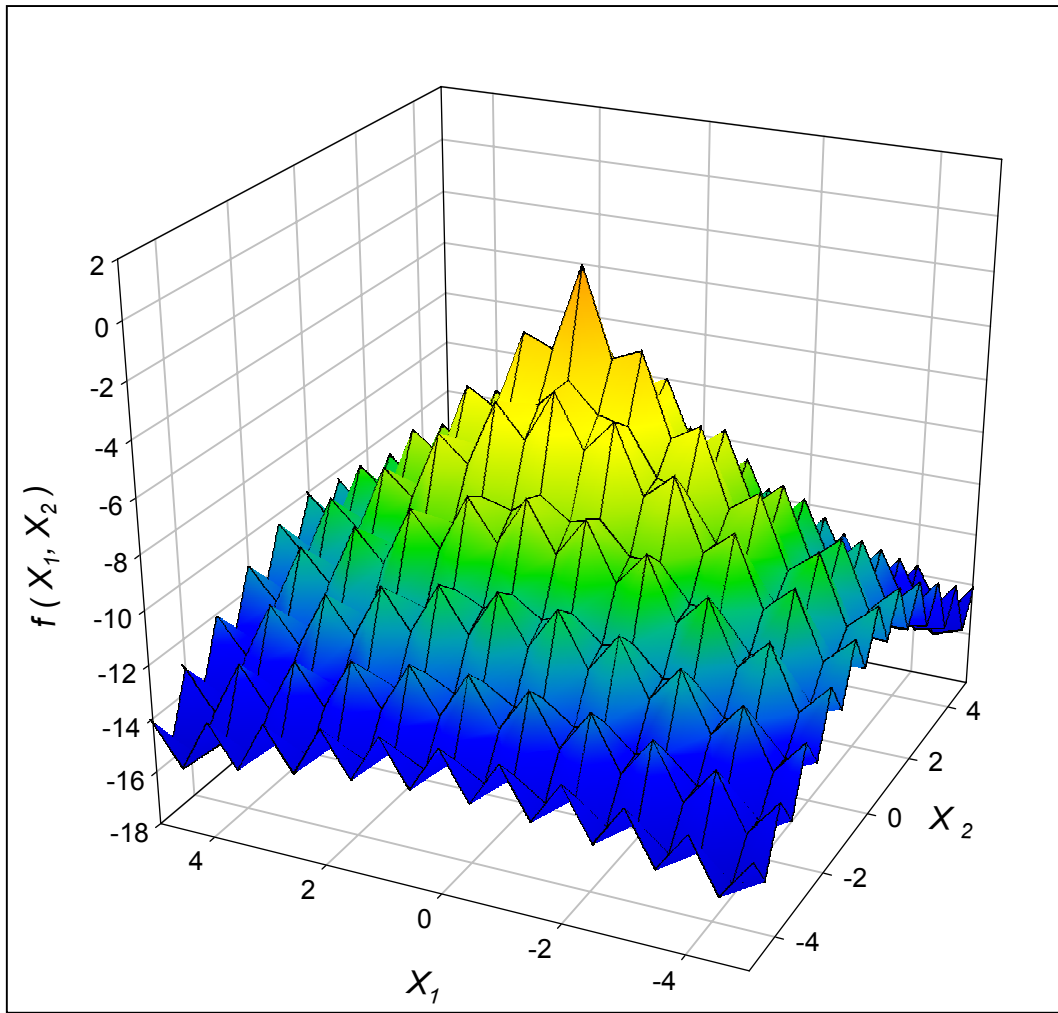


FIGURE 3.8 Rough surface of the test function.

The modified Ackley's function is expressed as:

$$y = c_1 \cdot \exp\left(-c_2 \sqrt{\frac{1}{2} \sum_{j=1}^2 x_j^2}\right) + \exp\left(\frac{1}{2} \sum_{j=1}^2 \cos(c_3 \cdot x_j)\right) - c_1 - e \quad 3.20$$

$$-5 \leq x_j \leq 5, \quad j = 1, 2$$

where

$c_1 = 20$, $c_2 = 0.2$, and $c_3 = 2\pi$. The known global maximum to modified Ackley's function is $f(x_1 = 0, x_2 = 0) = 0.0$.

3.4.1.2 The VisualGene solution

A relatively simple genetic algorithm with two-variable scheme to handle variables x_1 and x_2 is set up in VisualGene. **Table 3.2** summarizes the options used in VisualGene to solve the test function.

Figure 3.9 depicts parts of the calculation results obtained from VisualGene. The required chromosome length to accommodate both variables is calculated to be 30. After 154 iterations or $154 \times 5 = 770$ function evaluations, the optimum value is calculated by VisualGene in agreement with the known global maximum of the function to be:

$$f(x_1 = -15.25900 \times 10^{-5}, x_2 = -15.25900 \times 10^{-5}) = 4.71305 \times 10^{-3} \quad 3.21$$

TABLE 3.2 Genetic parameters used in VisualGene to solve the modified Ackley's function.

Operator or option	Value
elitism	yes
crossover	single-point
niching	no
child per pair of parents	1
micro genetic operator	yes
population size	5
mutation rate	1.0 %
crossover probability	50.0 %

The above example demonstrates the accuracy and validity of VisualGene calculations. In this study, the validity of VisualGene solution has been checked in one more occasion by plotting the surface of the optimization function. This matter has been discussed in section 3.5.1.

```

***   Generation 1   *****

***   Binary Code           Param1       Param2       Fitness
  1  100011111101010001001000001101  61.846e-2   -35.897e-1  -12.556375779484
  2  000110010010001101001110011100  -40.182e-1  15.322e-1  -11.2190479487026
  3  110010101011001110110101011010  29.18e-1    35.434e-1  -12.9515048436411
  4  010100111000101110111111010001  -17.367e-1  37.359e-1  -12.6896280497179
  5  111011111100000100100111110101  43.655e-1   77.807e-2  -11.3474490754727

Average Values:                42.943e-2    12.e-1      -12.1528011394036

Average Function Value of Generation= -12.1528011394036
Maximum Function Value           = -11.2190479487026
Number of Crossovers              = 64

Elitist Reproduction on Individual 2

***   Generation 50   *****

***   Binary Code           Param1       Param2       Fitness
  1  011010001010011101110100000101  -91.205e-2  22.674e-1  -8.78016893841854
  2  01101011110011101100100010101  -78.509e-2  19.597e-1  -7.59603226838878
  3  011111111111010100000000010100  -16.785e-4  62.563e-4  -2.111837824469E-02
  4  011010011110011100000000000000  -86.322e-2  15.259e-5  -2.72695619737715
  5  01101111111010100010100000001  -62.67e-2   39.109e-2  -4.46420672812417

Average Values:                -63.775e-2   92.492e-2  -4.71769650211067

Average Function Value of Generation= -4.71769650211067
Maximum Function Value           = -2.111837824469E-02
Number of Crossovers              = 70

***   Generation 154   *****

***   Binary Code           Param1       Param2       Fitness
  1  011111111111111100000000000000  -15.259e-5  15.259e-5   4.713045772604E-03
  2  001111111111101100000000000010  -25.008e-1  76.296e-5  -7.67072159009562
  3  011111111111101000000000000000  -45.778e-5  15.259e-5   4.024190226276E-03
  4  001110111111111100000011001010  -26.565e-1  61.8e-3    -7.7768333393211
  5  001111111101110100000000000010  -25.054e-1  76.296e-5  -7.67956948117611

Average Values:                -15.327e-1   12.726e-3  -4.62367743491879

Average Function Value of Generation= -4.62367743491879
Maximum Function Value           = 4.713045772604E-03
Number of Crossovers              = 71

%%%%%% Restart micro-population at generation 154

Summary of Output:
Generation   Evaluations   Avg.Fitness   Best Fitness
  1           5           -1.2153e1     -1.1219e1
  2          10           -8.4004e0     -4.1094e0
  3          15           -7.1143e0     -4.0065e0

 153         765           -7.1018e0     4.0242e-3
 154         770           -4.6237e0     4.713e-3
 155         775           -8.0412e0     4.713e-3

```

FIGURE 3.9 Selected results of VisualGene program in optimizing the modified Ackley's function.

3.5 Case studies

Application of the developed tool in tackling various optimization problems of importance to petroleum production is the main theme in this chapter. It has been tried in this part to demonstrate the performance of the field simulator and its components in solving different optimization problems that are usually encountered in petroleum production. The case studies that are discussed in this section are meant to represent the solution methodology for different classes of optimization problems rather than addressing an isolated optimization study.

The selected case studies present a collection of optimization problems that include both topside facility and reservoir unit. These case studies are believed to present a balance between reservoir and topside optimization problems.

To optimize decision variables belonging to topside facility, the reservoir output data at a specific time-step is sent to the process simulator to describe the inlet boundary of the process. The optimization routine then optimizes the decision variables based on the present inlet situation. In optimizing decision variables such as well placement that belong to the reservoir unit, the whole system should be solved in each iteration that is performed by the optimization routine. The reason is that while decision variables belong to the reservoir unit, the value of the objective function is normally calculated by the process simulator. Therefore, optimization of the reservoir parameters is more complicated than optimization at process level.

It is important to notice that unlike many other optimization studies that use simplified models to describe the behavior of production system components, basically no assumptions or simplifications are required to perform the optimization using the developed tool. The actual reservoir model that predicts the performance of the reservoir unit and the real topside simulation models can be used in the optimization, making the study as realistic as possible. The use of the tool is probably limited only by the available computing resources.

3.5.1 Case study 1 – Optimization of separator pressures

The first optimization case study presented in this section is determination of the optimum combination of pressures in the first and the second separators that will maximize the molar flow of the liquid product from the topside facility. This study can be considered as a PVT optimization where it is sought to find the best sequence of pressures at separation train to generate the maximum liquid hydrocarbon molar flow rate. The optimum stage pressures are normally determined by carrying out PVT experiments on samples of the reservoir fluid and process facilities are designed to accommodate the PVT recommendations. However, the day-to-day operation of the topside facility or variations in the composition of the reservoir effluent may require new pressure setting in separation stages. In such cases, an online optimization of stage pressures seems to help operating the facilities at optimal or near-optimal condition. The optimization study presented here is meant to propose a method to perform real-time optimization on selected process parameters that dictate directly or indirectly the operation economy. Although the focus here is on separation stage pressures, any other parameter or combination of parameters can be optimized using this method in a similar way.

This is an unconstrained optimization problem where the maximum value of the objective function, i.e., the produced liquid molar flow rate, is sought without considering any equality or non-equality constraints. The combination of the first and second stage pressures in the permissible range that generates the maximum value of the objective function is accepted as the optimum solution.

The topside facility has been simulated by HYSYS and is the real process currently in use in a North Sea field. **Figure 3.10** depicts the PFD of the topside facility. The decision variables are pressure drop across valves *Valve_1* and *Valve_2*. The outlet pressures of these valves determine pressures in the first and the second separators denoted as *1st_sep* and *2nd_sep*, respectively. The objective function is the molar flow rate of stream *Liquid* that is the bottom product of the *Fractionator* column.

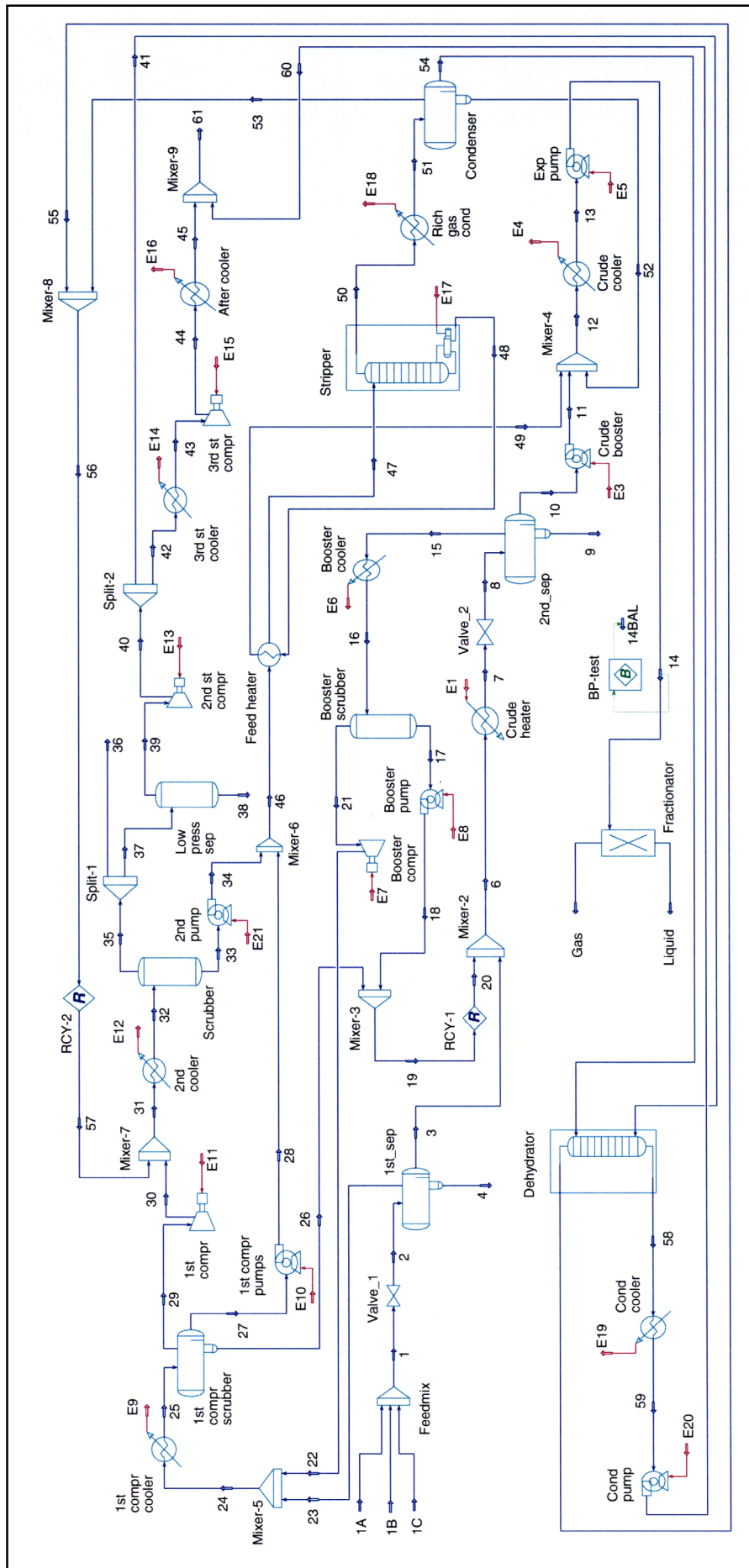


FIGURE 3.10 Process flow diagram of the topside facility of case study 1.

The optimization problem can then be stated as:

$$\begin{aligned} &\text{maximize} && \text{molar flow rate of stream } \mathit{Liquid} \\ &\text{subject to} && 0.0 \leq \Delta p \text{ across } \mathit{Valve_1} \leq 2300.0 \text{ kPa} \\ &&& 200.0 \leq \Delta p \text{ across } \mathit{Valve_2} \leq 400.0 \text{ kPa} \end{aligned}$$

The permissible ranges for variation of pressure drop across *Valve_1* and *Valve_2* have been selected so as to create a physically meaningful process condition. If HYSYS calculation for any unit operation fails to converge at any combination of these two pressure drop values, a zero value is assigned to the objective function. In this way, impossible process situations, or in other word, undefined regions of the search space are excluded.

Unlike the previous example, there is no explicit analytical expression for use in VisualGene to determine the value of objective function. This value is in fact calculated by HYSYS as one of the properties of the stream *Liquid*. VisualGene program was therefore modified to communicate with HYSYS through Automation using the methods explained in Chapter 2. At each iteration, VisualGene sends the randomly generated pressure drop values for first and second valves to HYSYS, commands HYSYS to perform the calculations, inquires back the calculated molar flow value of stream *Liquid*, and uses this value for fitness calculation.

Figure 3.11 shows selected parts of the VisualGene output. The required chromosome length is 30 and the genetic parameters indicated on **Table 3.2** were used. As is depicted on **Figure 3.11**, VisualGene finds the optimum combination of pressure drop values and accordingly, maximum value of the molar flow rate at generation 70. The maximum molar flow rate is determined to be 5785.4888 kmole/hr, corresponding to a pressure drop of 16.874×10^2 kPa across the first valve and a pressure drop of 200.05 kPa across the second valve. Accordingly, the first and the second stage separator pressures are determined as 812.60 kPa and 239.95 kPa, respectively.

The surface of the objective function was obtained by rigorous simulation in order to verify the validity of VisualGene calculations. The surface of the objective function is shown on **Figure 3.12**. The calculated maximum function value in the same range of decision variables was determined to be 5785.469, which is in acceptable agreement with results of VisualGene calculation.

As mentioned earlier, in this example, the molar flow rate of the liquid hydrocarbon product of the *Fractionator* has increased to 5785.5 kmole/hr. As can be seen from **Figure 3.12**, a wrong combination of separation pressures can reduce the production rate to values as low as 5711.8 kmole/hr. The revenue increase from this near-worst situation to the calculated optimal production corresponds to nearly 1.8 million dollars a month. This example clearly demonstrates the importance of carrying out such real-time optimization studies on existing facilities to determine the optimum production condition.

The optimization method presented above is also applicable if more than two separation stages exist in a process. By increasing the chromosome length, more decision variables can be accommodated in the search routine and optimum pressures for all separation stages can be determined in a similar way.

The performance of the VisualGene in maximization of the modified Ackley's function and the optimization study demonstrated above is believed to prove the validity of VisualGene calculations in the rest of this study.

```

*** Generation 1 *****

*** Binary Code          Param1      Param2      Fitness
1  100011111101010001001000001101  12.922e2    22.821e1    5778.47752927429
2  000110010010001101001110011100  22.581e1    33.064e1    5744.53950826907
3  110010101011001110110101011010  18.211e2    37.087e1    5739.22457413762
4  010100111000101110111111010001  75.057e1    37.472e1    5728.94780931634
5  111011111100000100100111110101  21.541e2    31.556e1    5720.49283149979

Average Values:                12.488e2    32.4e1      5742.33645049942

Average Function Value of Generation= 5742.33645049942
Maximum Function Value         = 5778.47752927429
Number of Crossovers           = 64

*** Generation 50 *****

*** Binary Code          Param1      Param2      Fitness
1  101110111010001000000001001000  16.858e2    20.044e1    5785.41535075684
2  111100101011010101001100110100  21.806e2    33.001e1    0
3  111110010100001110100111111111  22.395e2    36.562e1    0
4  110000111010010011111000011101  17.578e2    29.705e1    5766.06571527417
5  110011000100110110000010101100  18.355e2    35.105e1    5748.4875795475

Average Values:                19.398e2    30.884e1    3459.9937291157

Average Function Value of Generation= 3459.9937291157
Maximum Function Value         = 5785.41535075684
Number of Crossovers           = 73

Elitist Reproduction on Individual 4

*** Generation 70 *****

*** Binary Code          Param1      Param2      Fitness
1  101100011010111000001000001000  15.964e2    20.317e1    5784.7465524825
2  101110111010111000000011000000  16.862e2    20.117e1    5785.29706154383
3  101100010010111000001011001000  15.919e2    20.435e1    5784.52577110809
4  101110111010111000000001001000  16.862e2    20.044e1    5785.41862810087
5  101110111100111000000000001000  16.874e2    20.005e1    5785.48883475613

Average Values:                16.496e2    20.184e1    5785.09536959828

Average Function Value of Generation= 5785.09536959828
Maximum Function Value         = 5785.48883475613
Number of Crossovers           = 73

Elitist Reproduction on Individual 3

%%%%%% Restart micro-population at generation 70

Summary of Output:
Generation  Evaluations  Avg.Fitness  Best Fitness
1            5            5.7423e3    5.7785e3
2           10            5.7536e3    5.7785e3

69           345            5.7842e3    5.7854e3
70           350            5.7851e3    5.7855e3
71           355            4.5978e3    5.7855e3

```

FIGURE 3.11 Selected VisualGene output from case study 1.

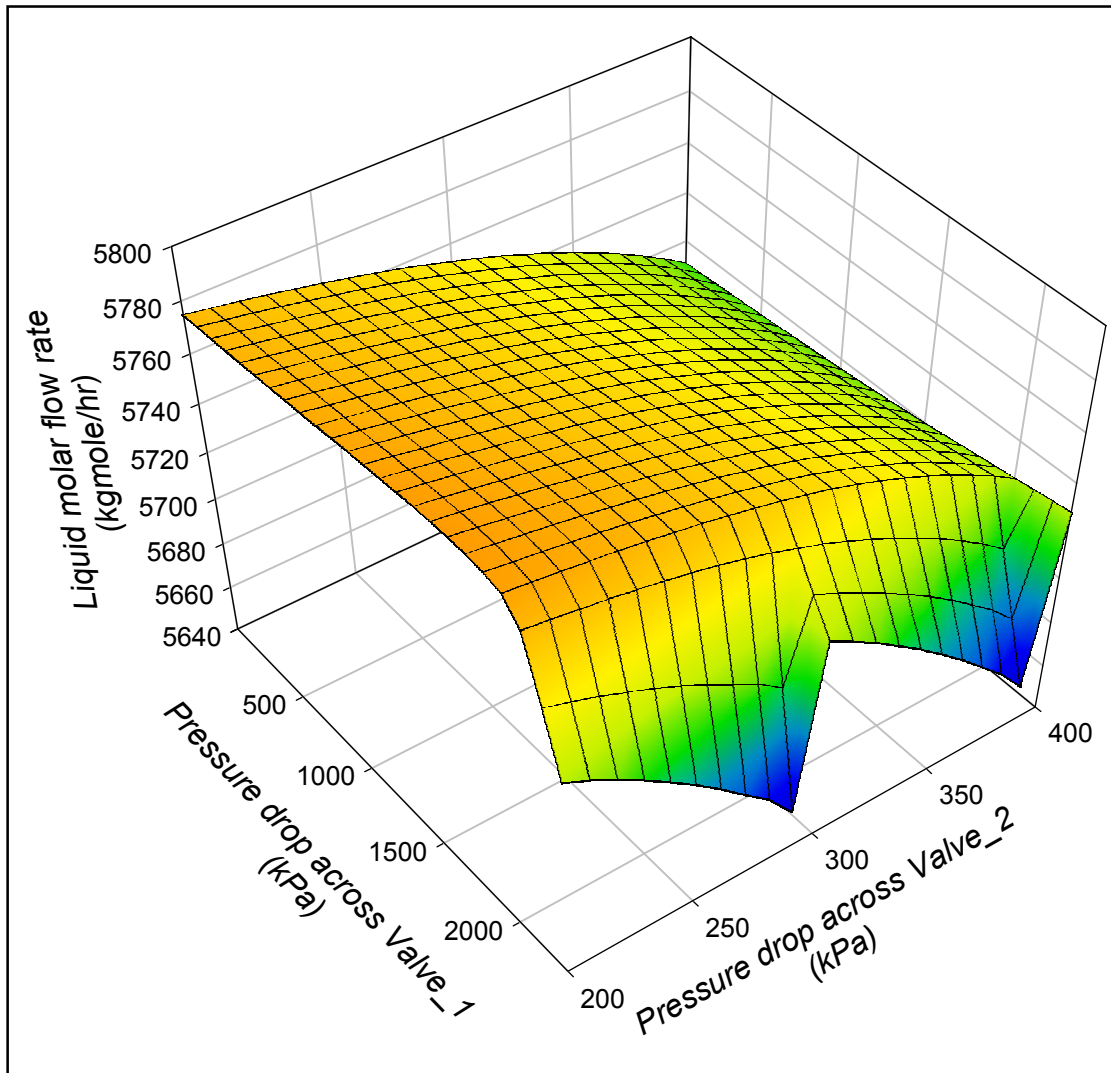


FIGURE 3.12 Surface of the objective function of case study 1, created by rigorous simulation.

3.5.2 Case study 2 – Tying satellite wells into existing facilities

Producing into existing production facilities normally is the only option to develop marginal fields. The main problem in using existing processing facilities is their limited capacity in processing the imposed extra load. If the facility is determined to be capable of handling the new load, then optimum allocation of the well streams in the system is the next question.

The second case study that is presented in this section is about optimum allocation of tie-in wells into existing facilities. More specifically, it is

sought to optimize allocation of n well streams between m separators without changing any process parameters in order to maximize the production from the processing unit.

The strategy that is taken to solve this problem using VisualGene is different from conventional numerical optimizations performed by VisualGene in previous examples. The problem here is basically to change the configuration of the processing unit in order to find the optimal solution. Each configuration creates a different value of the objective function and the search is for the arrangement that creates the optimum value of the objective function. In this case study, it has been tried to formulate a general approach to search for optimum solution in such kind of optimization problems.

This methodology is developed through this case study by optimizing the location of ten well streams that produce a marginal field between first and second separators of an existing processing unit. The topside facility presented in case study 1 is used again. The simulation of this process unit is complicated by iterative calculations of *Stripper* and *Dehydrator* columns, and trial-and-error calculation of recycle operations *RCY-1* and *RCY-2*. This case study that involves modification of PDF of this process is therefore believed to represent a realistic, and hard to solve optimization problem.

There are 2^{10} different ways that these ten streams can be allocated between first and second separators, therefore, random search for the optimum arrangement is out of question.

In this case study, the corresponding locations for each stream are problem variables. Therefore, there are ten variables in this problem the values of which will randomly be generated as either zeros or ones. A zero value means that the corresponding stream will be connected to the first separator and a value of one means that the corresponding stream will be located to the second separator. If three separators were involved, random numbers of zero, one and two could be generated to allocate streams between first, second, or third separator and etc. Therefore, the proposed method is general from this point of view. The objective function in this study is again the molar flow rate of stream *Liquid*.

Table 3.3 indicates the genetic parameters used in VisualGene to solve this problem. In this study, the length of the required chromosome to accommodate all ten variables is calculated to be 155. **Figure 3.13** depicts parts of the VisualGene output for this problem. Because of space limitations, only parts of the chromosomes have been shown. As can be seen from this figure, VisualGene finds the optimum arrangement after only 8 iterations or 40 function evaluations. The optimum arrangement corresponds to a molar flow rate of 30609.595 kmole/hr for stream *Liquid*, obtained by connecting first and sixth streams to the second separator and all the rest to the first separator. This is shown in the first chromosome of generation 8.

VisualGene surprisingly finds very fast the optimum solution in this case study. This speed can probably be attributed to the shape of the objective function surface near the optimum point. Genetic algorithms converge very fast to the optimal solution if the surface is more convex in vicinity of the optimal point.

Beyond the practical applications that such optimization can have, it demonstrates the power of genetic algorithms in optimizing configuration of the process unit. This example demonstrates how configuration modification can be performed programmatically through Automation. This widens the range of possible applications of the presented methodology to a great extent and leads to numerous interesting optimization studies that can be performed using the developed method. **Figure 3.14** shows the optimized PFD of the processing unit.

TABLE 3.3 Genetic parameters used in case study 2.

Operator or option	Value
elitism	yes
crossover	single-point
niching	yes
child per pair of parents	1
micro genetic operator	yes
population size	5
mutation rate	1.0 %
crossover probability	50.0 %

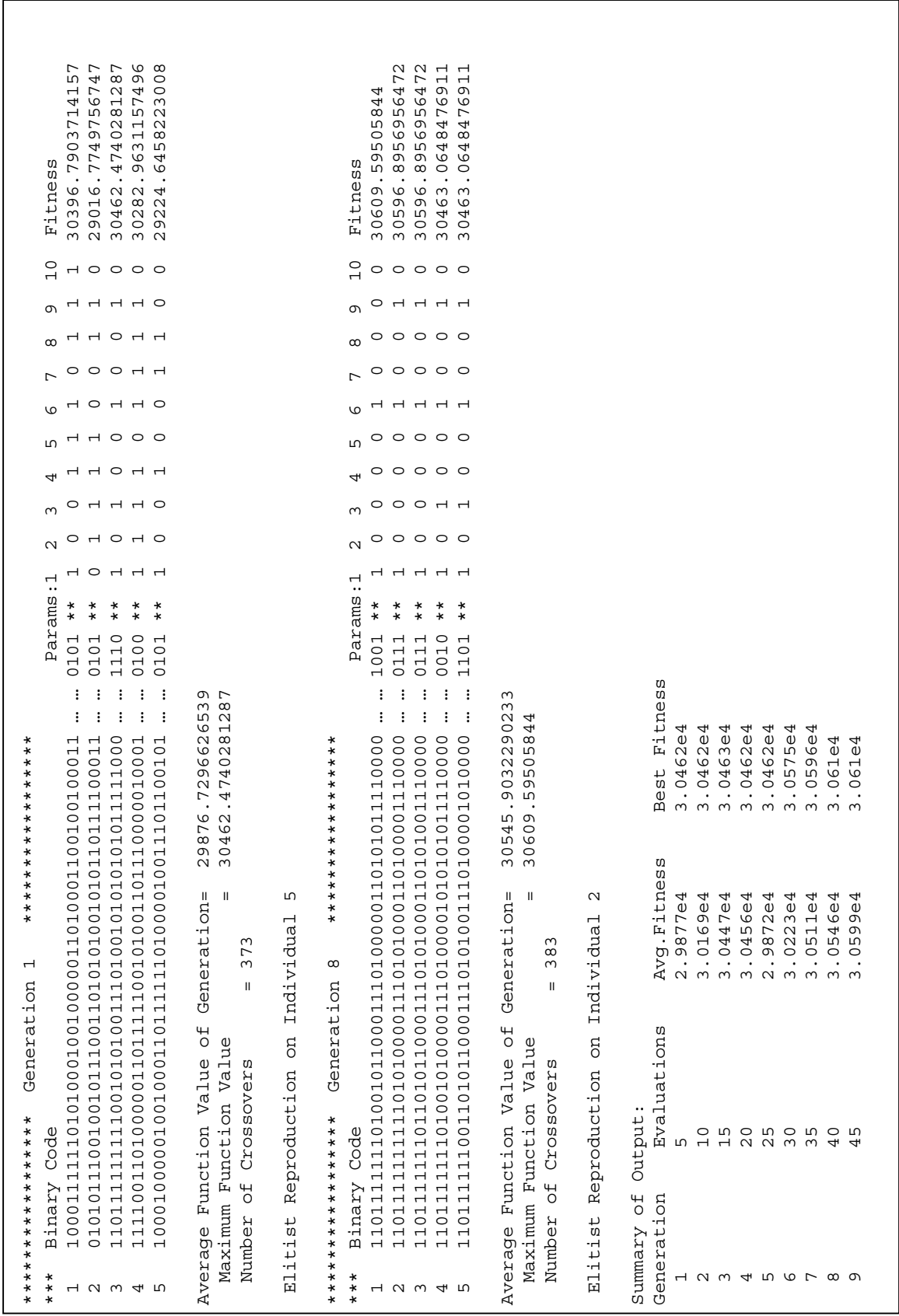


FIGURE 3.13 Selected VisualGene output from case study 2.

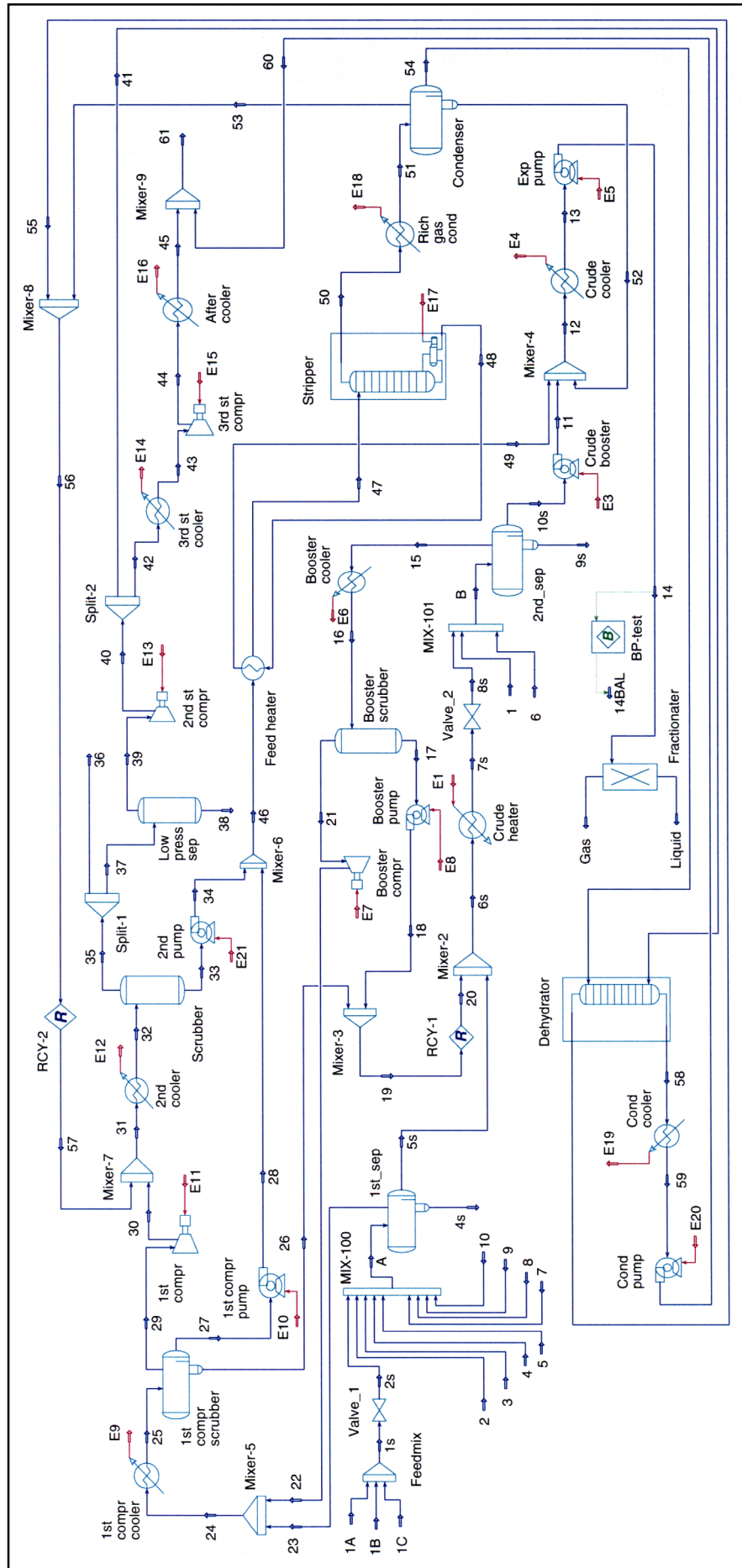


FIGURE 3.14 Optimized process flow diagram of the process unit of case study 2.

3.5.3 Case study 3 – Optimal well location

In complex reservoir systems, finding the optimal well location is far from being a trivial practice. In these reservoirs, there are such a huge number of interrelated and influencing parameters that the job of deciding on the best well location becomes a near impossibility. Normally, trial and error approaches are the only way out in such situations. After the reservoir geology has been established through interpretation of seismic data and exploration well logs, a number of possible well locations are usually obtained. The usual practice is then to run several simulation studies in these areas where sand quality and saturation is promising in order to establish the locations that result in maximum production from the wells.

Obviously, the random nature of such studies is considerably reduced through incorporating available information regarding regions of higher saturation or permeability. However, when the number of promising well locations exceeds a certain limit, it is extremely difficult to decide on the best candidate cases that need to be executed in order to find the optimum well location at a reasonably low simulation time. Therefore, it is necessary to have a systematic method to search for the optimum solution. This will certainly reduce the overall simulation time and hopefully find the optimum or near-optimal solution.

In this case study, performance of the developed tool together with VisualGene is demonstrated in finding the optimum well location in a realistic situation. The problem is to find the optimal well location in a small high pressure, gas condensate reservoir in the North Sea. The reservoir consists of a gas condensate bearing formation and a mostly water bearing formation. A thorough evaluation of the reservoir has been performed by the operator utilizing all relevant well data, reservoir parameters and fluid properties. Both depletion and pressure maintenance were investigated as potential exploitation strategies. The study concluded that the production of the resources by depletion from one well is the most robust and commercially attractive scenario. Finding the optimal well location is complicated because of the complex reservoir geometry. The reservoir is faulted with throws of approximately 20 to 80 m, containing a rich gas condensate fluid. The average zone permeabilities are in the range of 0.2 to 330 mD. This includes highly permeable streaks. The reservoir can be characterized as heterogeneous, with some restrictions to vertical and horizontal flows. Reservoir internal communication and the bulk rock

volume constitute the major reservoir uncertainties for the development of the reservoir under study. Several cases have already been simulated with wells located in various places in the reservoir. The performance of VisualGene will be demonstrated in this case study by determining the optimum location for a single vertical well. The productivity of the optimal well has to be compared with that of other promising locations to validate the optimization results.

An Eclipse 300 case with $30 \times 55 \times 24$ cells is used to model the reservoir. **Figure 3.15** depicts the reservoir geometry.

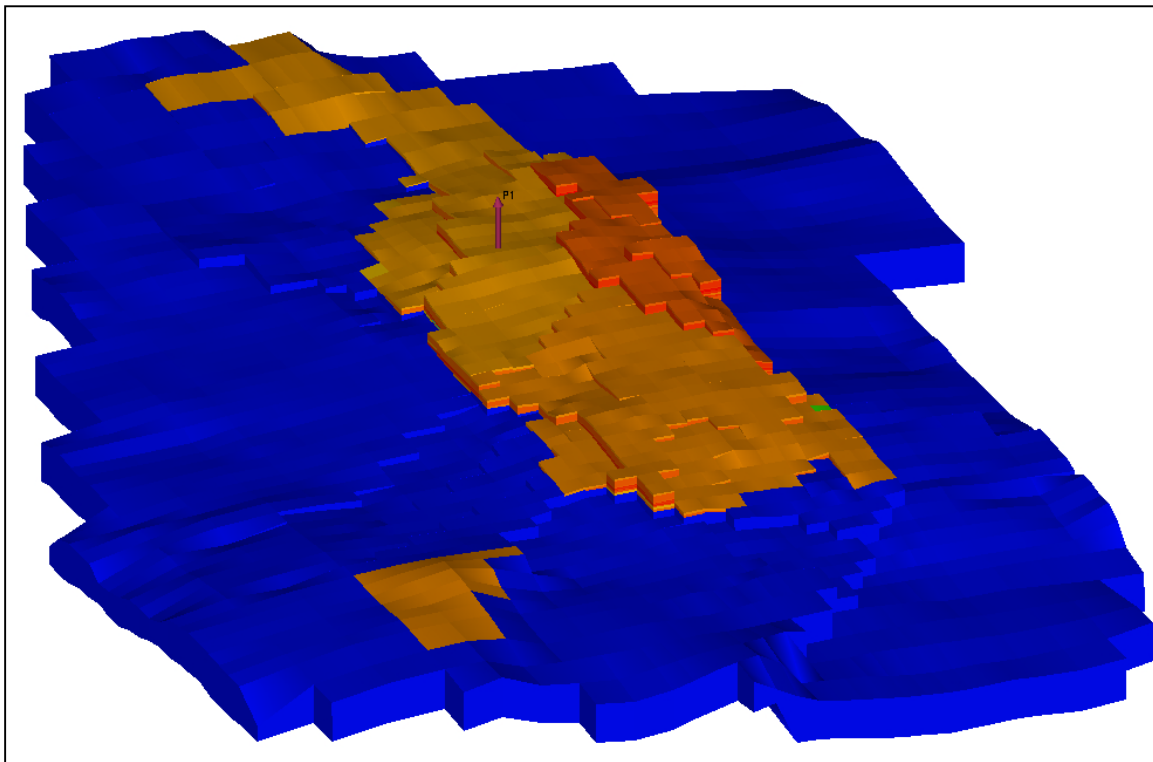


FIGURE 3.15 Initial gas saturation map of reservoir in case study 3.

Location of each well in Eclipse data file is determined by specifying $i, j, k1$, and, $k2$ parameters. These four numbers constitute the decision variables of the problem and the random location of a well can be determined by random generation of these four parameters. In each generation, VisualGene randomly produces five sets of these four values to generate five random well locations. Parameter i can vary from 1 to 30, parameter j can vary from 1 to 55, and parameters $k1$ and $k2$ can each vary from 1 to 24.

At each iteration, the WELSPECS and COMPDAT keywords in the Eclipse data file are updated programmatically to include the most recent, randomly generated well locations for use in the next reservoir calculation. The objective function that is the cumulative oil production after a specific period of time is inquired by LinkControl program at the end of each reservoir calculation and is recorded in the VisualGene output file.

The required chromosome length to accommodate all four variables required to locate a well in this 3-D reservoir model is 60. The genetic operators used in this case study are those depicted on **Table 3.3**. The reservoir model is run for 100 days at each iteration and the cumulative surface volume of the produced oil is inquired and used as the objective function. VisualGene converges continuously towards the optimum value indicating a smooth surface of the optimization function near optimal point. The optimum well location is found at generation 38 or after 190 function evaluations. Selected parts of the VisualGene output have been depicted on **Figure 3.16**. The optimum well location is determined to be at $i = 17, j = 37, k1 = 5,$ and $k2 = 24$. This location has been depicted on **Figure 3.15**. As can be seen from this figure, VisualGene has found a location near the top of the reservoir dome and mid in the high gas saturation area. A cross-section of the reservoir at the found location has been shown on **Figure 3.17**. As depicted, VisualGene has continued well completion to the bottom of the reservoir at this location with a considerable length of the well completed in the water zone. This is obviously not a desired situation but since the negative consequence of unwanted water production is not included in the calculation of objective function, VisualGene does not try to avoid placing well in water zone. In other words, since water production has not been defined as a constraint in the optimization problem, no negative point is associated to high water production by VisualGene's calculations. The issue of constrained optimization has been addressed in the next section.

As shown on **Figure 3.16**, the maximum cumulative oil production after 100 days from the optimal well location is 332027.8142 Sm^3 . The maximum cumulative oil production after 100 days from a number of promising locations in the vicinity of the found location were calculated in order to verify the validity of VisualGene calculations. **Table 3.4** shows the simulation results for four such promising locations. The first location, found by VisualGene, has the third best cumulative oil production. The second location has a completion from the top of the reservoir and has the second best cumulative production. The third location has the highest

production and is probably the global optimum. Although it has been located completely in the gas column, the fourth location has the lowest cumulative production. Considering the small difference in cumulative production between the found location and locations 2 and 3, it can be concluded that VisualGene has found a near optimal solution in this case. If the iterations were allowed to continue for more generations, VisualGene would have converged to the optimal solution. But since the search surface is probably not very convex near the optimum point, it would take a number of iterations before VisualGene can converge to the global optimum.

TABLE 3.4 Comparison between cumulative oil production after 100 days from promising regions in the reservoir.

Location	Coordinates				Cumulative production, Sm ³
	<i>i</i>	<i>j</i>	<i>k1</i>	<i>k2</i>	
1.	17	37	5	24	3.3203 x 10 ⁵
2.	17	37	1	24	3.3266 x 10 ⁵
3.	20	26	2	15	3.3310 x 10 ⁵
4.	23	32	1	24	2.9272 x 10 ⁵

3.5.4 Remarks

The developed method can determine the optimal well location using commercial 3-D reservoir simulations. No assumptions or simplifications are needed in the reservoir model. The method can handle more than one well by increasing the chromosome length. The criterion for optimality in such cases can be the maximal group cumulative oil production.

The optimum location depends on many parameters and can vary from situation to situation. A location that is optimum for early production is not guaranteed to remain competitive in long-term production.

```

*** Generation 1 *****
*** Binary Code
1 100011111101000100100000110100011001001000110100111001110011100
2 11001010100110101010110100100110001011100010111011111010001
3 1110111110000010010011110101010111001001011100101010100
4 1010110111000110010001110111110101010000010111101010000
5 0010001100101101110110011101101111111111101111010110101

Maximum Function Value = 295970.054667746
Number of Crossovers = 139

*** Generation 23 *****
*** Binary Code
1 100011111100110011011101101001111000100011101011101000000
2 1010101110100110010111110010111110100010001011101000001
3 1001101110101000010101101001101001101011100011101100010
4 100101011011101000111101000010100000100111100011001100
5 10001100101001111100111100111100001001110000111001001001001

Maximum Function Value = 331246.853645952
Number of Crossovers = 144
Elitist Reproduction on Individual 1

*** Generation 38 *****
*** Binary Code
1 100011111111110010100111010110101010011010111111110101101
2 1000111111100010011010111100001011010000011101011101000100
3 10001111111100101010001110000101010000001111111101000100
4 01011111010011101010001110001000010010010100101110001000
5 100011111110011110100001111000110100000011110101110000100

Maximum Function Value = 332027.814193502
Number of Crossovers = 145
Elitist Reproduction on Individual 3

Summary of Output:
Generation Evaluations Avg.Fitness Best Fitness
1 5 5.9194e4 295970.054667746
19 95 8.9528e4 330059.045876579
20 100 1.9612e5 331246.853645952
37 185 6.6411e4 331533.487126932
38 190 1.5768e5 332027.814193502

```

FIGURE 3.16 Selected VisualGene output from case study 3.

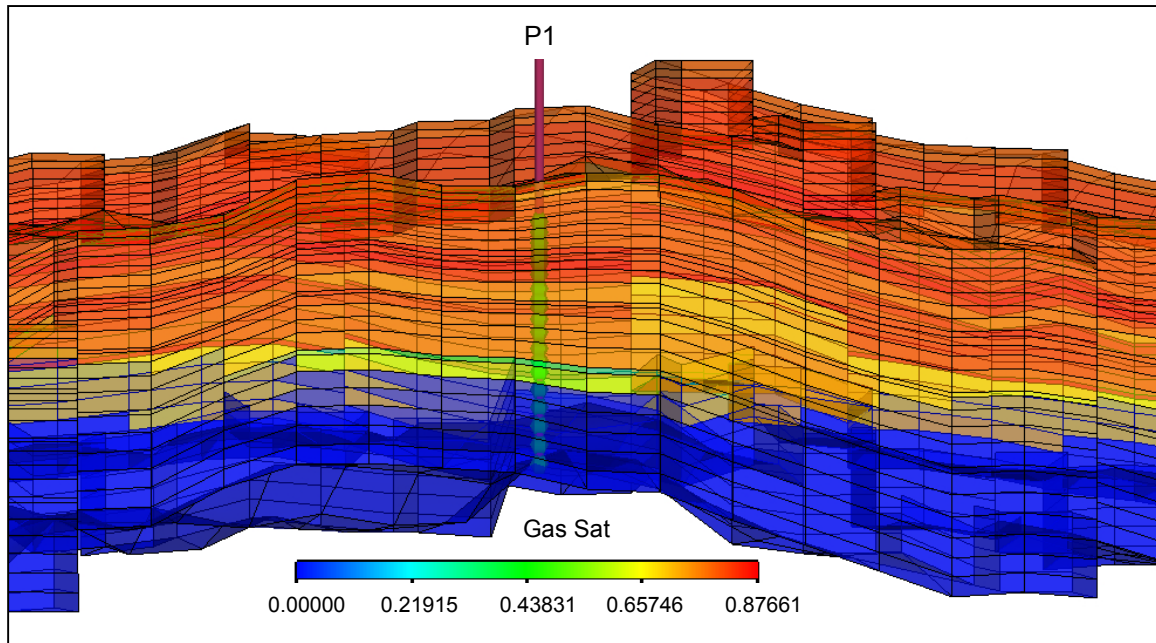


FIGURE 3.17 Optimal well location in reservoir of case study 3.

3.6 Constrained optimization

Thus far, in this discussion only application of genetic algorithms for optimizing unconstrained objective functions has been discussed. Many practical problems contain one or several equality or inequality constraints that must also be satisfied.

In this section, the incorporation of constraints into genetic algorithm search is considered. Case study 4 is designed to approach the optimization problem from a more realistic point of view and demonstrate the capability of the developed tool in solving constrained optimization problems.

3.6.1 Handling constraints

The general constrained optimization may be expressed as:

$$\begin{aligned} & \text{maximize} && f(x) && 3.22 \\ & \text{subject to} && g_i(x) \leq 0, && i = 1, 2, \dots, m_1 \\ & && h_i(x) = 0, && i = m_1 + 1, \dots, m (= m_1 + m_2) \\ & && x \in X \end{aligned}$$

where $f, g_1, g_2, \dots, g_{m_1}, h_{m_1+1}, h_{m_1+2}, \dots, h_m$ are real valued functions and x is an n -dimensional real vector with components x_1, x_2, \dots, x_n . The problem must be solved for the values of x_1, x_2, \dots, x_n that satisfy the restrictions and maximize the function f . Each of the $g_i(x)$ constraints is called an *inequality constraint* and each of the $h_i(x)$ constraints is called an *equality constraint*.

It would appear that constraints should pose no particular problem to the genetic search. A genetic algorithm generates a sequence of parameters to be tested using the system model, objective function, and constraints. The model is simply executed, the objective function is evaluated and checked to see if any constraint is violated. If not, the random solution is assigned the corresponding fitness value and is allowed to remain in the solution pool. If constraints are violated, the solution is infeasible and thus has no fitness. This procedure is fine except that many practical problems are highly constrained and finding a feasible solution is as difficult as finding the best one. As a result, it is necessary to obtain some information from the infeasible solution as well. This is usually done by degrading the fitness ranking of infeasible solutions in relation to the degree of constraint violation.

3.6.2 Available techniques

Several techniques have been proposed to handle constraints with genetic algorithms. (see for example Kim 1996, Michalewicz 1995, Myung 1995, and Orvosh 1994). Michalewicz et al. (1996) have reviewed nearly all available techniques in details. Gen and Chen (1997) classify the existing methods to handle constraints in genetic algorithms in this way:

Rejecting Strategy. In rejecting strategy all infeasible solutions are discarded during the evolutionary selection process. For simple search spaces, this method works efficiently enough but in many problems one should cross the infeasible region to reach the optimal solution. In such cases, the useful information from the infeasible region will not be available.

Repairing Strategy. This strategy involves repairing the infeasible chromosome in order to generate feasible one using some repairing procedures. While the method works satisfactorily with most optimization problems (Liepins, 1990), there is no general repair mechanism that works for different problems and therefore the repairing mechanism is to a great extent problem dependent. For each particular problem, a repair mechanism should be developed. In some problems, the process of repairing infeasible chromosomes can be as complex as solving the problem itself.

Modifying Genetic Operator Strategy. Instead of modifying the chromosome, some problem-specific genetic operators can be developed to deal with infeasibility in the chromosomes population. This method has proven to be a successful approach (Michalewicz, 1996) and the only disadvantage is the problem-dependence of the genetic operators.

Penalty Strategy. All the above mentioned methods try to avoid inclusion of infeasible solutions into the chromosome pool. For highly constrained problems, infeasible regions can constitute the majority of the search space and in such cases it is difficult to find feasible solutions. Glover et al. (1989) have suggested that moving through infeasible region yields more rapid convergence and produces better final results. The penalizing strategy is a technique developed to use the information obtained from the infeasible region in the genetic search. The penalty strategy has been used in this study to handle the infeasible solutions.

In a penalty method, the constrained problem is transformed into an unconstrained problem by penalizing the infeasible solutions. This cost or penalty is included in the objective function evaluation.

Considering the original constrained problem in maximizing form:

$$\text{maximize } f(x) \quad 3.23$$

$$\text{subject to } h_i(x) \geq 0, \quad i = 1, 2, \dots, m$$

where x is an m vector

one can transform this to the unconstrained form:

$$\text{maximize } f(x) + \sum_{i=1}^m r_i \Phi[h_i(x)] \quad 3.24$$

where Φ is the penalty function,
 r_i is the penalty coefficient

A number of alternatives exist for the penalty function (see for example, Joines 1994, Michalewicz 1995, Smith 1993, and Gen 1996). In this study, a method proposed by Goldberg and modified by Homaifar et al. (1994) is used to define the penalty function. The penalty function is constructed with the penalty coefficient r and the square of the violation of the constraint as follows:

$$\begin{cases} \Phi[h_i(x)] = 0 & \text{if } x \text{ is feasible} \\ \Phi[h_i(x)] = h_i^2(x) & \text{otherwise} \end{cases} \quad 3.25$$

r_i in general is a variable penalty coefficient for the i th constraint. For each constraint, several levels of violation are created. r_i values vary according to the level of violation. Determining the level of violation for each constraint and choosing suitable values of r_i is not so trivial and is highly problem-dependent.

Michalewicz (1995) demonstrated that if the penalty coefficients are moderate, the algorithm may converge to an infeasible solution and if the

penalty coefficients are too large the method approaches the rejecting strategy.

3.6.3 Case study 4 – Optimal well location with water production constraint

This case study involves finding the optimal well location in the presence of one constraint. The objective function is the well cumulative oil production with well cumulative water production as constraint. The optimum well is therefore supposed to have the maximum cumulative oil production after a specific period of time while the water production from the well should be minimum.

As mentioned, the penalty strategy is used in this case study to handle the constraint. The objective function is therefore modified at each iteration to consider the penalty associated to water production. Using **Equation 3.25**, the square root of the cumulative water production from the well is subtracted from the cumulative oil production to calculate the penalized fitness of individual chromosomes. In this way, a variable penalty is considered for each chromosome. The more the extent of the constraint violation, the more the value of the assigned penalty. Since there is only one constraint in this problem, there is no need to weigh the amount of the penalty assigned to each constraint and therefore the penalty coefficient r is set equal to one.

The inclusion of the penalty term into the objective function increases the convexity of the search surface and causes the genetic search to converge to the global optimum with fewer function evaluations.

The reservoir unit presented in case study 3 is used again. The random well location is determined using the same technique explained before. The only difference is that LinkControl inquires the well cumulative water production at each iteration so that VisualGene can calculate the amount of the assigned penalty using **Equation 3.25**.

The objective function is calculated after 365 days of reservoir production in order to study the performance of the VisualGene at longer-term production. **Figure 3.18** depicts the selected output from VisualGene for case study 4. As can be seen from this figure, program converges to the optimal solution already at generation 7 or after 35 function evaluations. The optimum

location is found to be at $i = 20$, $j = 34$, $k1 = 8$, and $k2 = 18$. Maximum cumulative oil and water production at this location after 365 days are calculated to be 1.0950×10^6 and 11.815 Sm^3 , respectively. **Figure 3.19** depicts the optimal well location.

No increase in calculated maximum value of the objective function is observed up to generation 15 where all 5 chromosomes in the generation reproduce the same well location. This means that the global maximum is probably reached and further calculation is therefore not necessary. As was mentioned earlier, this fast convergence can be attributed to the increased convexity of the search surface because of inclusion of penalty function.

Figure 3.20 shows the reservoir cross section at the found well location. As expected, VisualGene has tried to avoid water zone in order to minimize the extent of the penalty imposed on the calculated fitness because of violation of the water production constraint.

Manual inspection of other promising locations reveals that the found position is indeed the global optimum. **Table 3.5** shows the simulation results for the found optimal location and three other promising locations. As can be seen from this table, the found location has the maximum cumulative oil production and the minimum cumulative water production. There is no sensible increase in production after one year if the well is completed from the top of the reservoir, instead water production increases slightly.

TABLE 3.5 Comparison between cumulative oil production after 365 days from promising regions of the reservoir model.

Location	Coordinates				Cumulative production, Sm^3	
	i	j	$k1$	$k2$	Oil	Water
1.	20	34	8	18	1.0950×10^6	1.1815×10^1
2.	20	34	1	18	1.0950×10^6	1.7204×10^1
3.	20	26	2	15	1.0950×10^6	2.0530×10^2
4.	23	32	1	24	1.0206×10^6	2.8189×10^1

```

*** Generation 1 *****
*** Binary Code
1 100011111010100010010000001101000110010010000110100111001110011100
2 11001010101001101101010110100101001110001011101111010001
3 1110111110000010010011110101010111001010111001101010100
4 101011011100011001000111011110110101010000010111101010000
5 0010001100101101110110011101110111111111111110110110101

Maximum Function Value = 1094019.12802534
Number of Crossovers = 139

*** Generation 4 *****
*** Binary Code
1 1010110111000110010001010111110110101010000010111101010000
2 10101110110001100100011101111011010101000010111101010000
3 101011011100011001100111011101101010100010010111101010000
4 101011010110010001110001011111011010101000010101111011000
5 10101110010001100110011111111011010101010010111111010000

Maximum Function Value = 1094306.85794061
Number of Crossovers = 156

*** Generation 7 *****
*** Binary Code
1 11101110100000010010011101010110101010101010110011101010000
2 101011011100011101001010111111110101010001010111101010000
3 10100101011001100110111011111011100010001001001100010000
4 101011011100011001100111011101101010100010010111101010000
5 1110110111000110011001111111011010101000101011110110110000

Maximum Function Value = 1094860.40216333
Number of Crossovers = 153

Summary of Output:
Generation Evaluations Avg.Fitness Best Fitness
1 5 -3.9635e12 1094019.12802534
2 10 -3.0985e13 1094019.12802534
3 15 -1.5055e12 1094306.85794061
4 20 -4.9364e11 1094306.85794061
5 25 -2.6552e12 1094306.85794061
6 30 -1.2969e12 1094306.85794061
7 35 -1.3015e12 1094860.40216333

i j k1 k2 Fitness
17 9 3 16 -3568254138819.47
24 47 9 21 -6569935960109.60
28 32 9 22 -8563028626897.99
21 32 10 17 1094019.12802534
5 40 12 18 -1116037661520.32

i j k1 k2 Fitness
21 32 10 17 1094019.12802534
21 32 10 17 1094019.12802534
21 33 10 17 1094306.85794061
21 7 9 17 -2468182556604.79
21 33 10 17 1094306.85794061

i j k1 k2 Fitness
28 17 12 20 -1369111505682.01
21 47 10 23 -526108986108.983
20 34 8 18 1094860.40216333
21 33 10 17 1094306.85794061
28 33 10 17 -4612125063419.75

Maximum Function Value = 1094860.40216333
Number of Crossovers = 153

Summary of Output:
Generation Evaluations Avg.Fitness Best Fitness
1 5 -3.9635e12 1094019.12802534
2 10 -3.0985e13 1094019.12802534
3 15 -1.5055e12 1094306.85794061
4 20 -4.9364e11 1094306.85794061
5 25 -2.6552e12 1094306.85794061
6 30 -1.2969e12 1094306.85794061
7 35 -1.3015e12 1094860.40216333

```

FIGURE 3.18 Selected VisualGene output from case study 4.

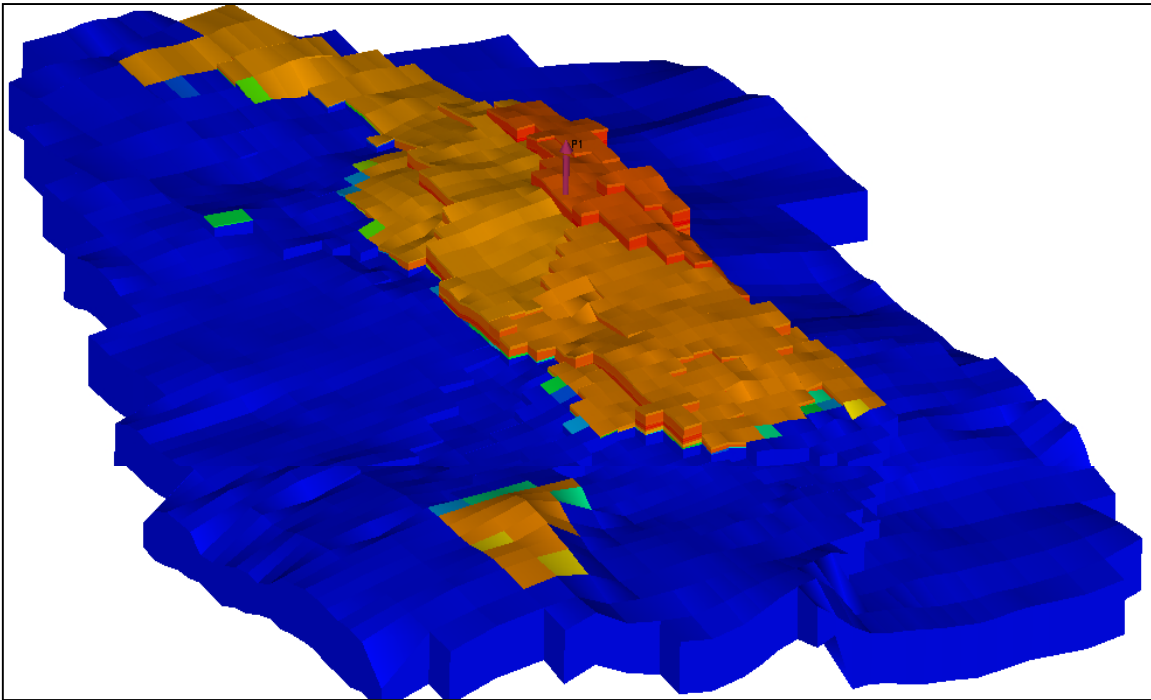


FIGURE 3.19 Initial gas saturation map and optimal well location for reservoir in case study 4.

3.6.4 Handling constraints and discontinuous surfaces in VisualGene

A simple and practical method to handle constraints was successfully tested in case study 4. This was achieved by minor modification of objective function calculation in VisualGene. This demonstrates yet another advantage in application of genetic algorithms for solving optimization problems. Handling constraints is not that trivial in other methods.

The optimization functions that were solved in case studies 3 and 4 represent the class of discontinuous functions. According to the reservoir simulator rules, only nonzero integer values are acceptable for variables $i, j, k1$, and $k2$. The function value is not defined for zero or non-integer values and the surface of the objective function is therefore discontinuous. This matter becomes clearer by considering the situation shown on **Figure 3.21a**. The depicted surface represented by red points belongs to a reservoir parameter for a typical well completion in Z direction from $k1$ to $k2$. This surface can for example represent the cumulative oil production after a specific time. The function value belonging to each pair of i and j values then represent the production from a well located at $i, j, k1$, and $k2$.

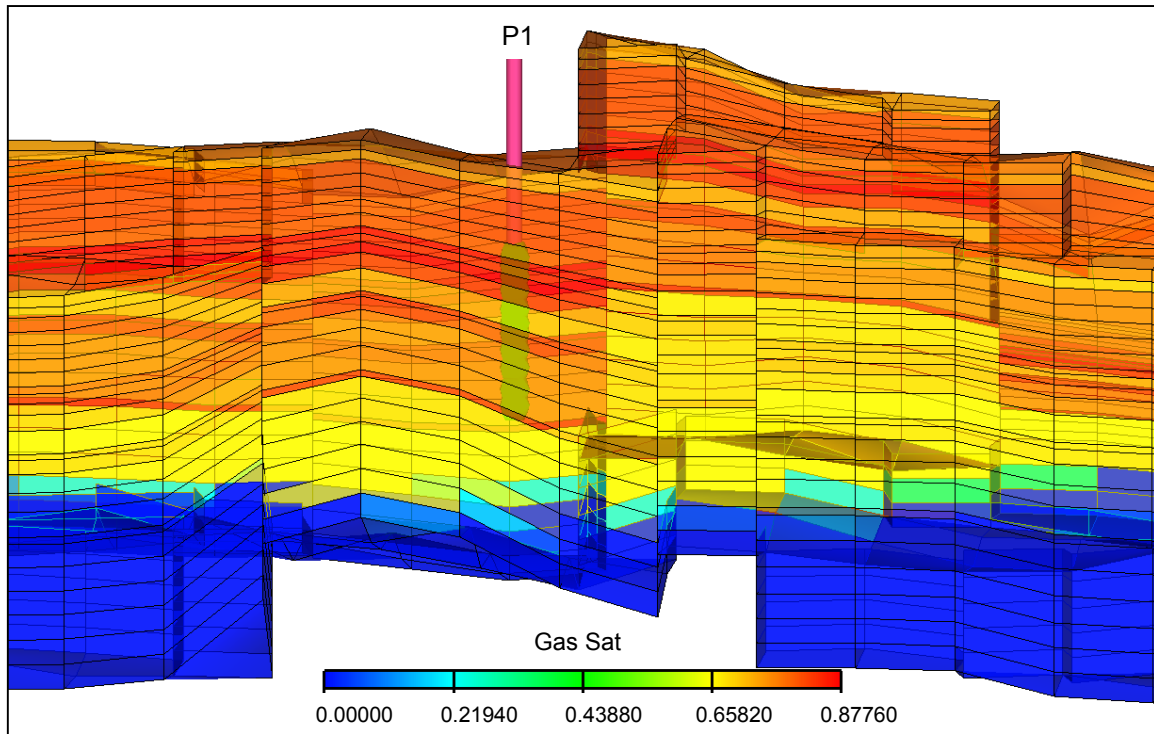


FIGURE 3.20 Optimal well location in reservoir of case study 4.

On the other hand, the internal calculations of VisualGene require double precision real values for decision variables i , j , $k1$, and $k2$. Enormous calculation time is wasted if VisualGene is allowed to search unpromising areas presented by real valued decision variables where fitness is zero. To resolve this problem, the randomly generated real valued decision variables are rounded to the nearest integer before being sent to the reservoir simulator for calculation of the objective function. In this way, a single function value is assigned to all decision variables that fall in the range of:

$$\frac{i + (i - 1)}{2} < i \leq \frac{i + (i + 1)}{2} \quad 3.26$$

$$\frac{j + (j - 1)}{2} < j \leq \frac{j + (j + 1)}{2}$$

$$\frac{k_n + (k_n - 1)}{2} < k_n \leq \frac{k_n + (k_n + 1)}{2} \quad n = 1, 2$$

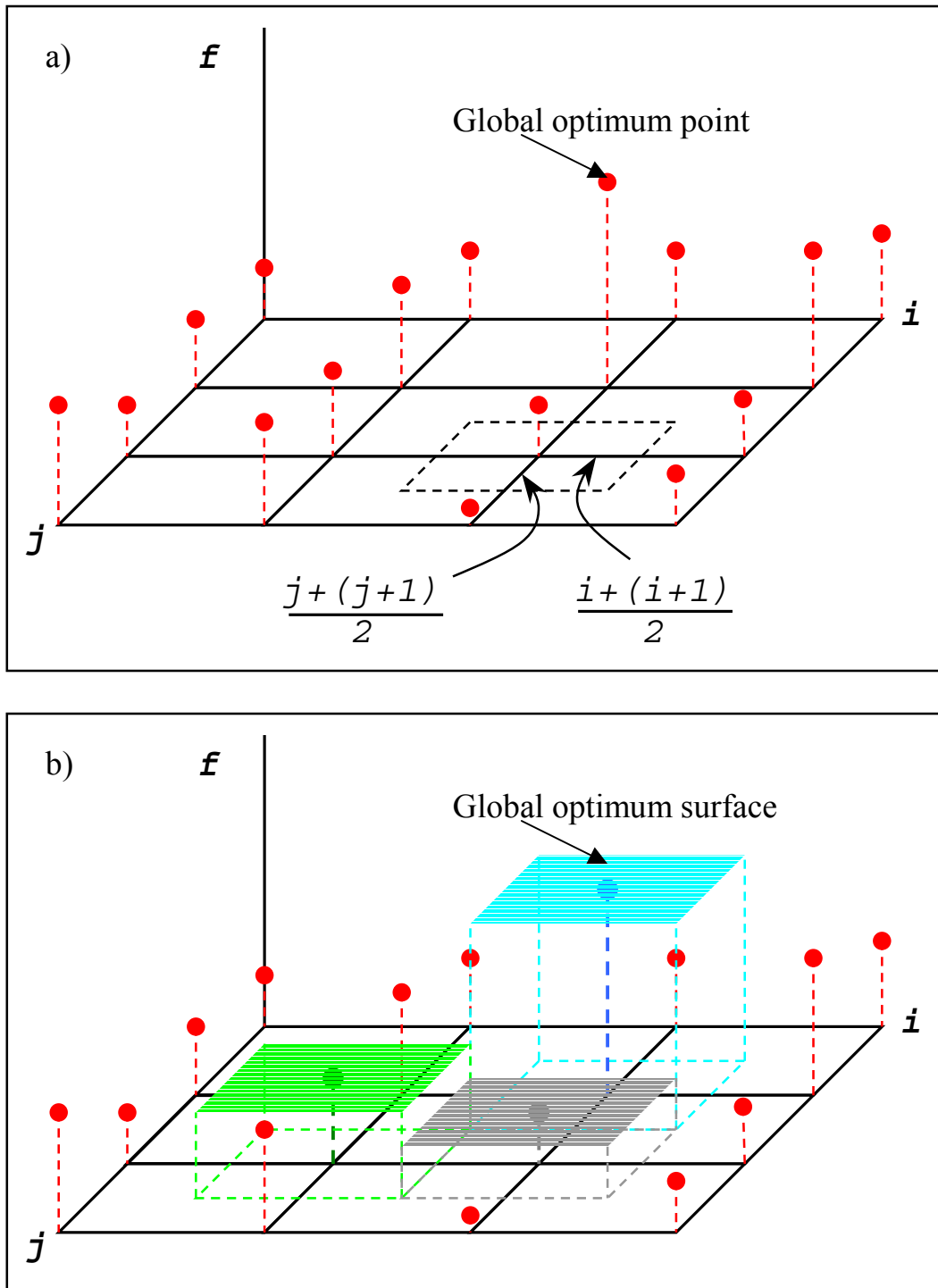


FIGURE 3.21 a) The discontinuous surface of objective function presented by a collection of points. b) Modified discontinuous surface for use in VisualGene calculations.

This will change the surface of the objective function from a collection of discontinuous points to a collection of discontinuous surfaces as shown on **Figure 3.21b**. As the result of this modification, precision of internal calculations of VisualGene remains intact by using double precision real values while numerical value of the global optimum is not jeopardized. Obviously, because of nature of genetic search, the discontinuity of the new surface causes no problem to VisualGene calculations.

3.6.5 Remarks

Based on the results obtained from optimization studies presented in this chapter, the following general observations can be made regarding expected performance of a search routine that is used in integrated simulators:

1. Optimization programs that are employed in integrated systems should not increase CPU requirements to a level that an integrated solution becomes infeasible.
2. The optimization routine is expected to converge rapidly to the optimum solution, especially when solution to some system components involves heavy iterative calculations.

Internal calculation of VisualGene is fast and only a fraction of total CPU time is used by VisualGene calculations. The total simulation time required by genetic operations of VisualGene is several orders of magnitude smaller than that of other system components.

HYSYS calculations are reasonably fast and simulations are usually finished in matter of minutes even if solution involves iterative calculations. Network solution is reached typically after few minutes depending on the network settings when Eclipse results are transferred to the network. After the network converges at the first time-step, convergence in subsequent steps is reached in matter of seconds.

Reservoir simulation time varies considerably depending on size of the model and type of the required simulation. While simulation times in range of minutes can be expected from a simple black oil model, it can take several days to solve a full-field compositional simulation.

Table 3.6 shows the CPU usage for each simulator in solving the optimization studies performed in this chapter. As can be seen from this table, reservoir simulator is usually the critical component.

TABLE 3.6 CPU consumption* for each simulator in optimization studies.

Problem	VisualGene	HYSYS	Eclipse	Total usage
Ackley function	0.801	-	-	0.801
Case Study 1	3.030	549.865	203.000**	755.895
Case Study 2	11.200	73.631	203.000	287.831
Case Study 3	47.290	-	38570.0	38617.290
Case Study 4	8.711	-	7105.00	7113.711

* Reported in seconds on an 850 Mhz Intel® Pentium III® processor.

** Single simulation up to a given time-step.

From the comparison of CPU consumptions, it can be concluded that HYSYS simulation is reasonably fast and no especial treatment seems necessary to reduce its simulation time in the integrated system. An option however exists in HYSYS to approximate simulation cases with a Neural Network. The Neural approximation reportedly reduces the simulation time significantly (HYSYS documentation, 2000).

VisualGene's internal calculation is fast but the required number of iterations to reach the optimum solution can cause problems in optimization problems that involve repeated reservoir simulations. In such situations, an integrated solution is feasible only if simulation time and number of iterations can be kept reasonably low. Three approaches are recommended to achieve a faster convergence in such cases:

1. Performance of genetic search increases considerably by selecting proper population size and genetic operators (Sen, 1992). However, finding the optimum genetic parameters requires some trial calculations that can take equally long time if no prior experience with the optimization problem exists.
2. It is in general possible to reduce the reservoir simulation time through a verity of available methods. For instance, the FLUX option in Eclipse simulator can be utilized to divide the reservoir into several smaller parts

(Eclipse documentation, 2002). A parallel search can then be set up to search for optimum solution on each part of the reservoir. The global optimum can be found by comparing the search results on each separated part of the reservoir.

3. In a recent study, Singh reports development of methodologies to reduce the reservoir simulation time (Singh, 2002). Using the proposed methods, a faster simulation is achieved by:
 - a. Reducing the number of cells in a simulation to reduce the simulation time without decreasing the accuracy.
 - b. Reducing number of components in fluid characterization to speed up the simulation without compromising the accuracy.
 - c. Modifying fluid descriptions so that black oil simulation can be used in some situations instead of time-consuming compositional simulation.

Using the mentioned methods or similar technologies that are already available, it is in general possible to reduce the total simulation time in an integrated system so that the application of the integrated system remains feasible. The integration method is however required not to restrict functionalities of participating components so that such methods can be implemented.

The expensive calculations discussed above originate from the probabilistic nature of the search strategy that has been used in the integrated system. Reported research results indicate that the use of such strategies is inevitable in handling the class of optimization problems presented in this chapter (see for example Fujii 1994, and Palke 1997).

According to available research, two frontrunners in this field are simulated annealing and genetic algorithm search strategies (Sen 1992, Thornton 1994, and David 1987). Without applying simulated annealing, it is impossible to comment on its performance in handling the presented simulation problems. However, based on the documented studies that compare the performances of simulated annealing and genetic algorithms and the experience of applying genetic search in this study, the following conclusions can be made

regarding the feasibility of applying genetic search in integrated systems as a viable alternative to simulated annealing:

1. A genetic algorithm routine that is capable of utilizing the power of genetic search and uses optimum combination of genetic parameters and operators, can reach the optimal solution rapidly and consequently can compete with simulated annealing search in this respect (Sen, 1992).
2. Complex reservoir simulations favor optimization methods that have a potential for parallel processing (Schulze-Riegert, 2001). In genetic algorithms, entire population can be processed in parallel (Holland, 1992). Consequently, genetic search has comparable performance with modified simulated annealing (Sen, 1992)
3. Genetic algorithms can handle any kind of objective functions and any kind of constraints, defined on discrete, continuous or mixed search spaces (Gen, 1997).

Chapter 4

Production system analysis

Perhaps the ultimate goal of an integrated simulation system is to assist in the practice of field planning. The privilege of having access to real-time data from all subsystems results in flawless decision making. In an integrated environment, each subsystem models and solves its own share of the system and therefore solution to the full model does not become more difficult as the system is enlarged. In most field planning studies that do not use integrated simulation, simplifying assumptions are needed to make the system solvable. In such cases, the results lose their validity since the simplified models lose their resemblance to the actual system.

This chapter is designed to demonstrate how the integrated simulation system can be used as a field planning tool. By the use of a case study, development of a fictitious field is demonstrated using the field simulator.

4.1 Introduction

Case study 5, discussed in this chapter, analyzes different production scenarios to produce a generic high pressure, high temperature, deepwater field. This analysis assumes a predefined reservoir development plan that fixes the number of wells and their locations according to an already executed optimization study. A deepwater reservoir has been assumed here in order to identify critical parameters involved in the development of deep and ultra deepwater fields. The focus is on finding the best production scheme among a number of possible alternatives. The final comparison between the proposed scenarios is made based on comparative net present value of each alternative. Real-time reservoir data directly inquired from the reservoir simulator at predefined time-steps is transferred to piping and process simulators to size the required network and determine the expected gas and liquid production from the field. No simplifying assumption is made in the reservoir behavior, making the results as general as possible.

4.2 Reservoir unit

The field under study is a high pressure, high temperature, deepwater prospect consisting of five marginal reservoirs scattered over a distance between 5 to 50 km. Each reservoir unit is produced by three horizontal producers. There is a production manifold for each reservoir unit and a single pipeline transports effluent from each reservoir unit to a common production manifold where the field production is commingled and transported for processing. The field is produced through existing facilities, either offshore or on land. As a consequence, the processing parameters are fixed. The field layout is illustrated on **Figure 4.1**.

Reservoir units are characterized by thin oil column, active aquifer and initial gas cap. The reservoir behavior is described by a compositional model with 9 components. **Table 4.1** summarizes some basic reservoir data for reservoir units. The fluid characterization used in the Eclipse 300 model was developed using the techniques discussed in Chapter 2. The expected overall field production profile has been depicted on **Figure 4.2**. The field is produced at about 144600 bbl/d for a short plateau period of about six months. The production rate declines finally to about 25000 bbl/d that is considered as the economical limit. The expected variations in gas oil ratio and water-cut are depicted on **Figure 4.3**. An early water breakthrough is expected with water-cut increasing to more than 90 percent at the end of the second year of production. This early water breakthrough makes the job of the network design more demanding.

Most of the reported similar studies assume some special reservoir behavior in order to make the system solvable. The most commonly used simplifying assumptions are constant water-cut and GOR throughout the reservoir life. In some other instances, lookout tables are applied to approximate the expected reservoir behavior. No simplifying assumptions have been made regarding the reservoir behavior in this study. The same is true with simulations of network and topside facilities.

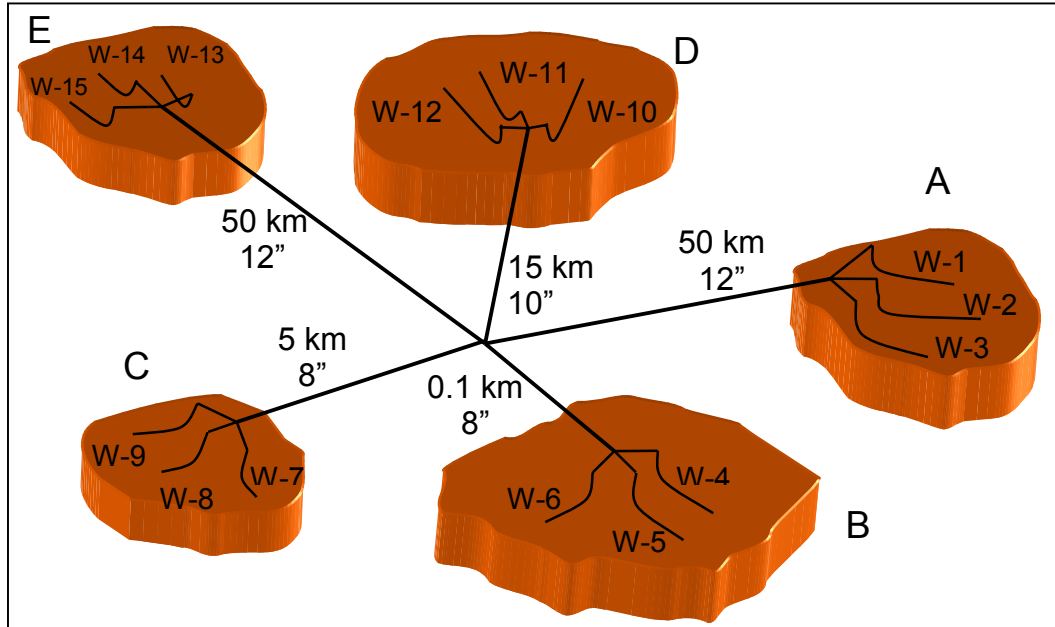


FIGURE 4.1 Layout of the generic field of case study 5.

TABLE 4.1 Basic reservoir data for the field under study.

Reservoir pressure, bar	290
Reservoir temperature, °C	200
Reservoir depth, m MSL	3000

Reservoir	A	B	C	D	E
GIIP, M Sm ³	1806.0265	1897.8584	1377.4779	1989.6903	1897.8584
OIIP, M Sm ³	8.3579	8.7829	6.3747	9.2079	8.7829

Component	Mole fraction	Property	Liquid	Gas
C1	0.634621	Density, kg/m ³	883.9060	0.8602
CO2	0.007519	Viscosity, cP	19.3875	1.0708 x 10 ⁻²
C2	0.070736	Molecular weight	327.1239	20.3034
C3	0.041570			
C4	0.017149			
C5	0.006412			
C6F3	0.012739			
F4	0.015883			
F5F6	0.193372			

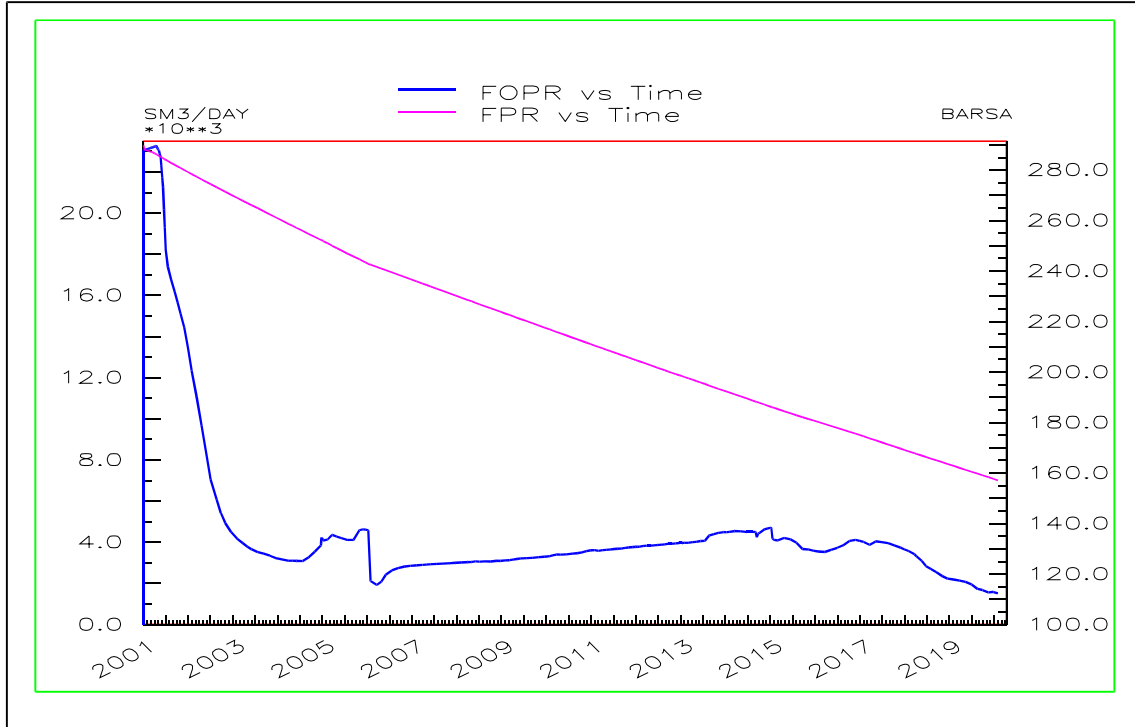


FIGURE 4.2 Expected overall production profile for the field under study.

4.3 Network

The production network consisting of 15 wells and 5 transfer lines has been simulated by HYSYS using the technique explained in Chapter 2. The reservoir data necessary to solve the network consists of well compositions, hydrocarbon molar flow rates, water production rates and reservoir pressure. These data are inquired by LinkControl at each time-step and sent to HYSYS to solve the network. LinkControl transforms water from a phase to a component as required by HYSYS. The vertical segments of the wells are simulated by HYSYS using the method explained in Chapter 2. The production network has been depicted on **Figure 4.4**.

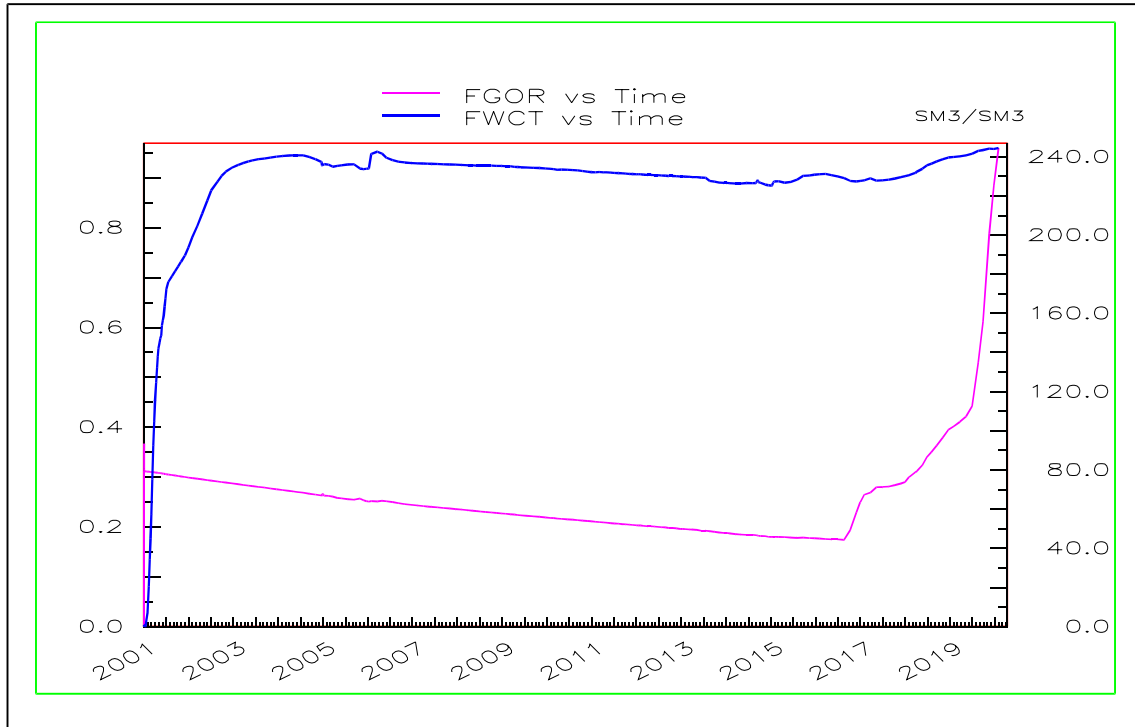


FIGURE 4.3 GOR and water-cut variations for the field under study.

Despite the complexity of the network and the fact that a multiphase flow situation exists in the network, the convergence of the network at the required precision is quite satisfactory. With current settings that require a precision of 0.01 bar in pressure calculations, the network converges at about 2 minutes. Simulation of the network at the same time involves sizing the pipes and designing the insulations. The network will converge only if all pipe segments have been correctly sized and properly insulated.

A network design is satisfactory only if it converges at early production period as well as when the reservoir has been depleted. An improperly designed network will not converge when the network inlet condition changes gradually as reservoir is depleted.

A comparison between the network explained in Chapter 2 and the network used in this case study reveals that the proposed technique is not restricted to simple networks with limited number of branches. The technique works acceptably in modeling complex converging networks with large number of branches and can handle multiphase flow situation.

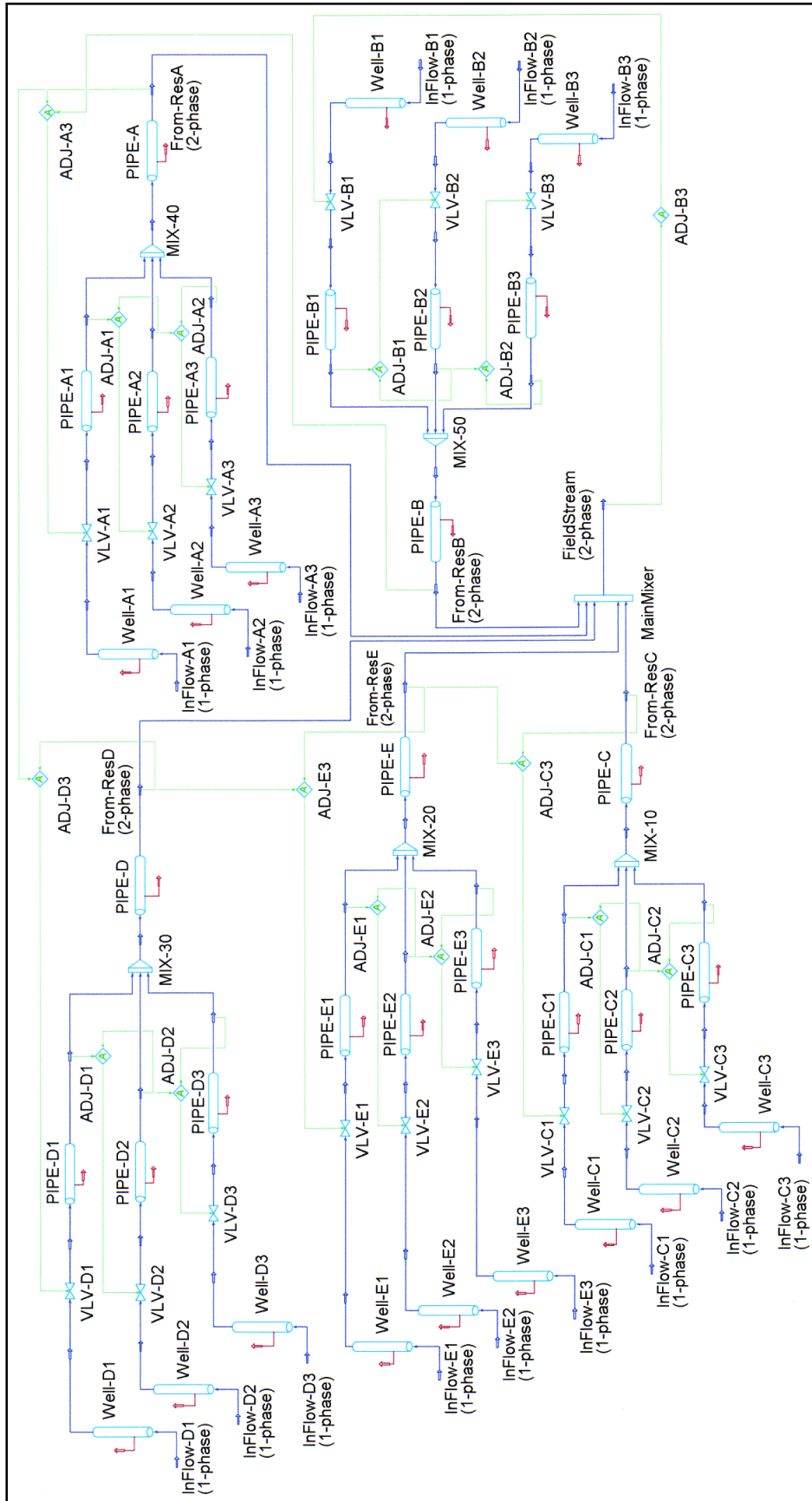


FIGURE 4.4 The production network simulated by HYSYS.

Since the network calculation involves a number of iterations at each time-step, it was decided to simulate the transfer line and process unit in a separate HYSYS simulation case in order to avoid unnecessary iterations.

The sub-sea network is assumed to operate at 5 °C with water speed of about 1 m/s. The same material is assumed in all pipe segments and the same type and thickness of insulation has been designed for all pipes.

4.4 Riser, processing plant and transfer line

Five different alternatives to transport and process the field effluent were considered. In the following sections each alternative is explained and discussed separately. The basic production parameters are fixed for these alternatives since the field is produced into existing facility. The fixed processing parameters are the number of separators and their corresponding pressures.

Separator pressures are set at 812.3, 239.3 and 101.3 bar, respectively and are optimized. The sub-sea separator however, operates at the network's delivery pressure. This pressure gradually reduces as the field is depleted.

4.4.1 First scenario

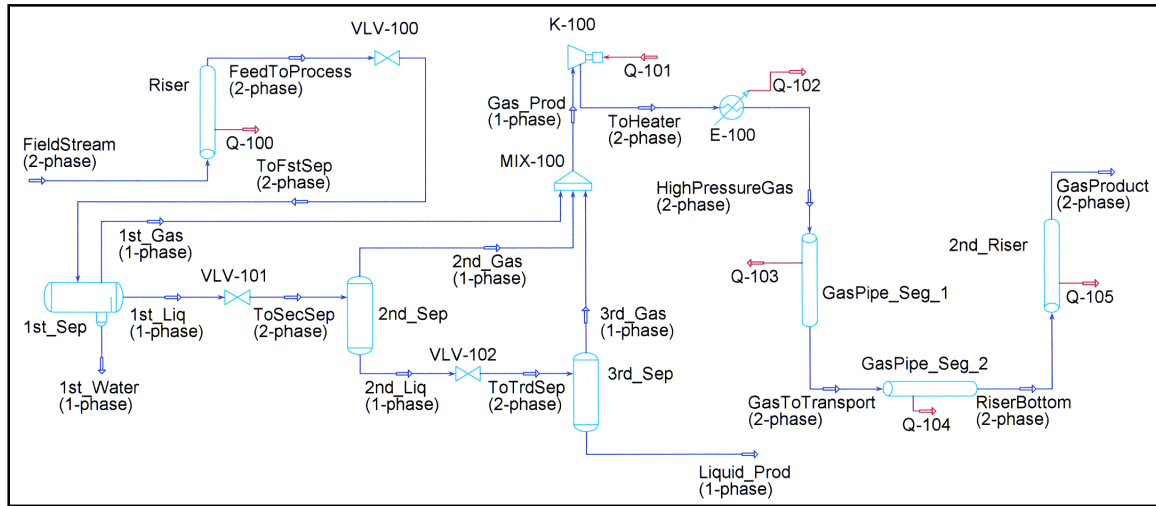


FIGURE 4.5 Schematic diagram of the first production alternative.

The first scenario is the most commonly used alternative to produce offshore fields. In this alternative, the commingled field stream is sent directly to an offshore processing plant through a riser mounted to the main manifold. The simplified processing plant consists of three separation stages. In the first separator, the produced water is separate. The liquid product is collected from the third separator and is transported by tanker. Gas product is compressed and transported to shore using a 15-inch, 100 km long transfer line. Important piping parameters are shown in **Table 4.2**.

TABLE 4.2 Basic data for the first alternative.

Pipe segment	Length m	ID in	OD in	Insulation k W/m K x 10 ⁻⁴	Insulation thick. m x 10 ⁻²	Roughness m x 10 ⁻⁵
Riser	1500	16	17	5.000	1.000	4.572
GasPipe_Seg_1	1500	16	17	5.000	1.000	4.572
GasPipe_Seg_2	100000	15	16	5.000	10.00	4.572
2nd_Riser	1500	16	17	5.000	1.000	4.572

4.4.2 Second scenario

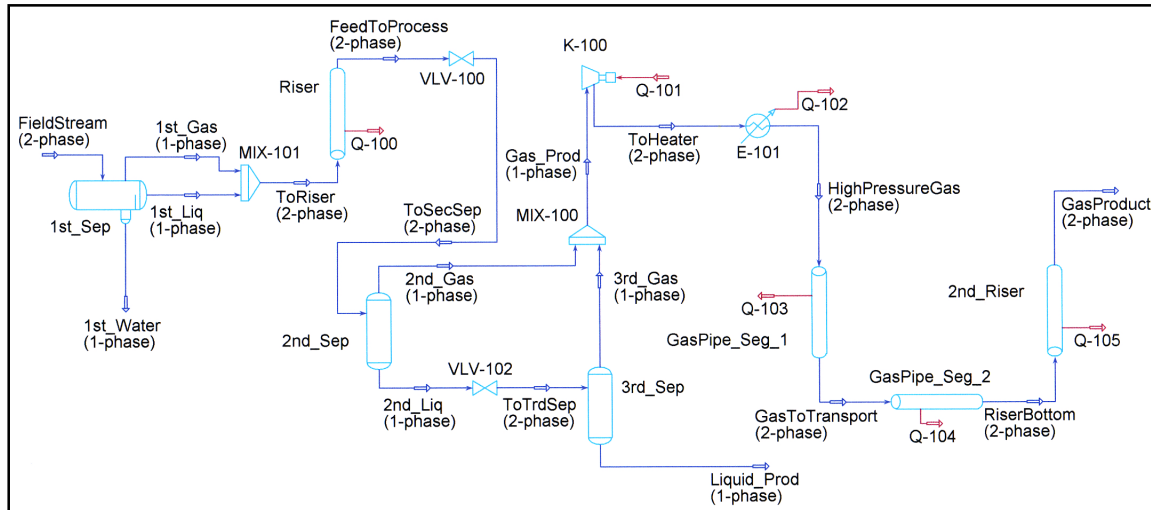


FIGURE 4.6 Schematic diagram of the second production alternative.

In the second alternative, a sub-sea separator installed at the seabed separates the produced water before the field effluent is sent to the riser. As will be discussed later, this method proves to be superior to the first alternative where no sub-sea processing has been planned.

In this case, the topside separation train has only two separation stages. The liquid product is once again collected from the third separator and transported by tanker and the gas product is compressed and sent to shore using a transfer line. Important parameters are reported in **Table 4.3**.

TABLE 4.3 Basic data for the second alternative.

Pipe segment	Length m	ID in	OD in	Insulation k W/m K x 10 ⁻⁴	Insulation thick. m x 10 ⁻²	Roughness m x 10 ⁻⁵
Riser	1500	16	17	5.000	1.000	4.572
GasPipe_Seg_1	1500	16	17	5.000	1.000	4.572
GasPipe_Seg_2	100000	15	16	5.000	10.00	4.572
2nd_Riser	1500	16	17	5.000	1.000	4.572

4.4.3 Third scenario

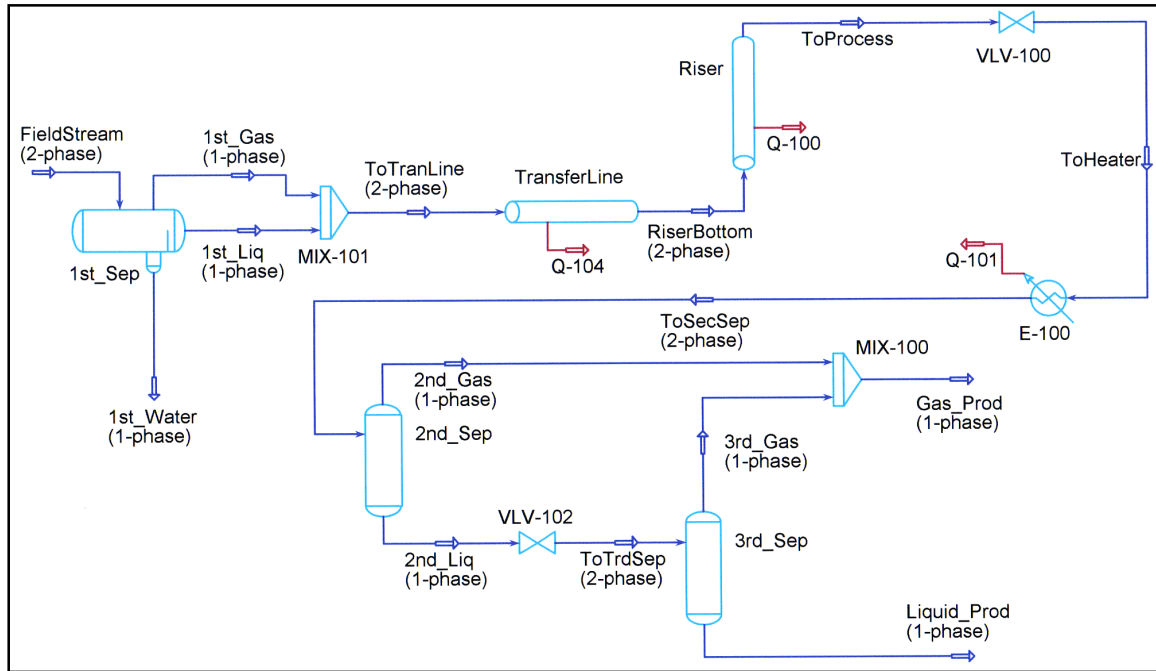


FIGURE 4.7 Schematic diagram of the third production alternative.

In this alternative, after separation of the produced water by the use of a sub-sea separator, the produced hydrocarbon phase is transported to shore for processing using a 22-inch transfer line. Similar to second alternative, there are only two separation stages at the topside facility. Important piping parameters have been reported in **Table 4.4**.

TABLE 4.4 Basic data for the third alternative.

Pipe segment	Length m	ID in	OD in	Insulation k W/m K x 10 ⁻⁴	Insulation thick. m x 10 ⁻²	Roughness m x 10 ⁻⁵
Transfer line	100000	22	23	5.000	10.00	4.572
Riser	1500	16	17	5.000	1.000	4.572

4.4.4 Fourth scenario

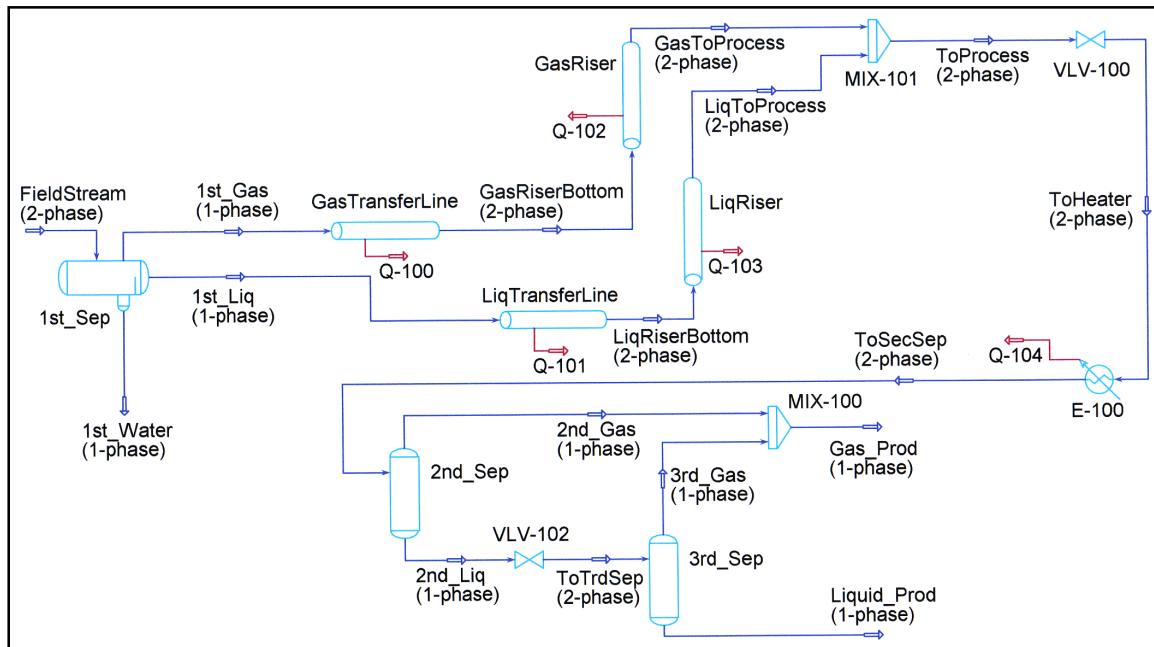


FIGURE 4.8 Schematic diagram of the fourth production alternative.

Fourth alternative presents a solution for production from fields with large production potential and has been studied here as a comparison. Separate transfer lines and risers have been provided for the produced hydrocarbon gas and liquid phases. A sub-sea separator is once again used to separate the water at the seabed. As has been discussed in the coming sections, this is one of the least attractive alternatives to produce the field under study. Important parameters have been reported in **Table 4.5**.

TABLE 4.5 Basic data for the fourth alternative.

Pipe segment	Length m	ID in	OD in	Insulation k W/m K x 10 ⁻⁴	Insulation thick. m x 10 ⁻²	Roughness m x 10 ⁻⁵
GasTransferLine	100000	15	16	5.000	10.00	4.572
GasRiser	1500	16	17	5.000	1.000	4.572
LiqTransferLine	100000	22	23	5.000	10.00	4.572
LiqRiser	1500	16	17	5.000	1.000	4.572

4.4.5 Fifth scenario

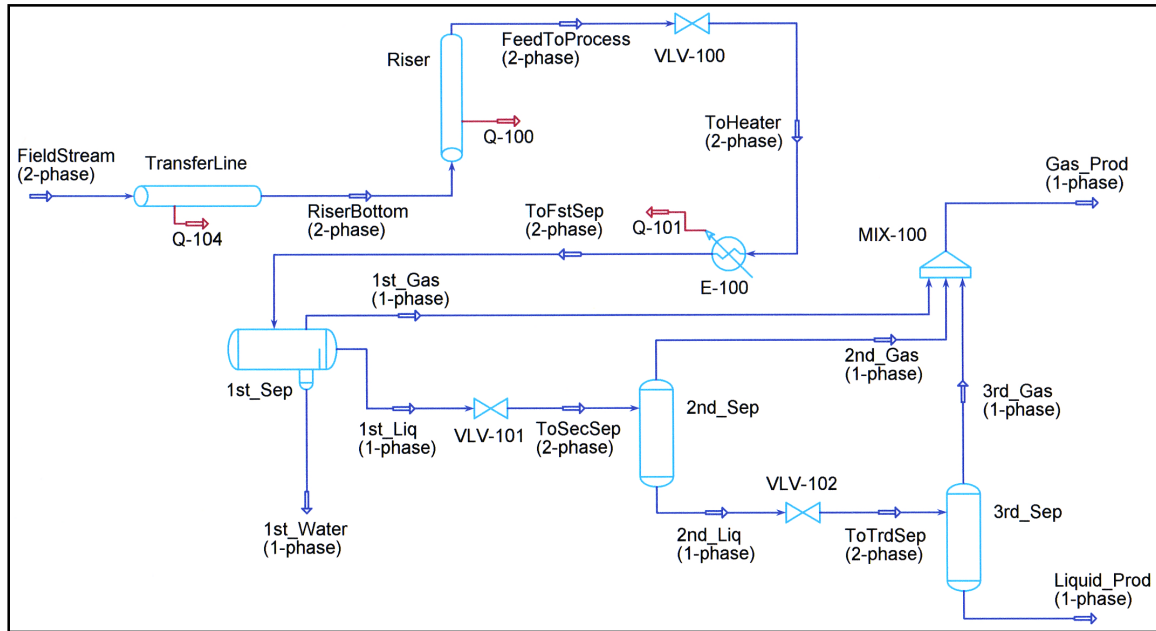


FIGURE 4.9 Schematic diagram of the fifth production alternative.

In this case, the reservoir effluent is transported to shore without any sub-sea separation or processing. A three-stage separation has been used at the processing facility to separate the produced phases. Important parameters have been reported in **Table 4.6**.

TABLE 4.6 Basic data for the fifth alternative.

Pipe segment	Length m	ID in	OD in	Insulation k W/m K x 10 ⁻⁴	Insulation thick. m x 10 ⁻²	Roughness m x 10 ⁻⁵
Transfer line	100000	22	23	5.000	10.00	4.572
Riser	1500	16	17	5.000	1.000	4.572

4.5 Comparison of alternatives

In comparing the alternatives it is important to notice that the configuration of processing facility and its key parameters have been kept constant in all alternatives. This includes the pressure values used in separation train. This simulates the situation when the field is produced into existing facilities. The inlet pressure and temperature of the transported gas is the same in all scenarios in order to make it possible to compare the alternatives. The sizes of the pipelines have been designed for each alternative depending on the size of the flow. Consequently, different inside diameters have been used in different alternatives.

Drilling horizontal producers near the water oil contact is the only way to produce the thin oil layers that exist in reservoir units. This will result in early water breakthrough with water-cut increasing to nearly 70 percent shortly after production starts. Despite the fact that the field under study is a high pressure field, nearly all alternatives require some sort of artificial lift after the field is produced for a period of time. The reason is considerable pressure drop in wells and risers because of huge amount of produced water.

First alternative needs pressure boost at the bottom of the riser after 180 days. This alternative is operational up to 5500 days, which is the economical life of the project. Second and third alternatives need a small pump to keep the required pressure in third separator after 4500 days. Second scenario is operational up to 5500 days. Third alternative is operational up to 4500 days. Because of two-phase flow situation at the riser bottom in this scenario, a multiphase pump is required to boost the pressure for production beyond 4500 days. However regarding the costs of installation and operation of a multiphase pump and the fact the field is already near the end of its life, the installation of a multiphase pump cannot be justified. In the fourth alternative, no gas is produced at seabed condition after 180 days. Therefore, the huge cost associated with construction of gas transfer line and riser is not justified. This scenario is operational up to 4500 days. Fifth alternative needs pressure boost at riser bottom after 180 days and is operational up to 5000 days.

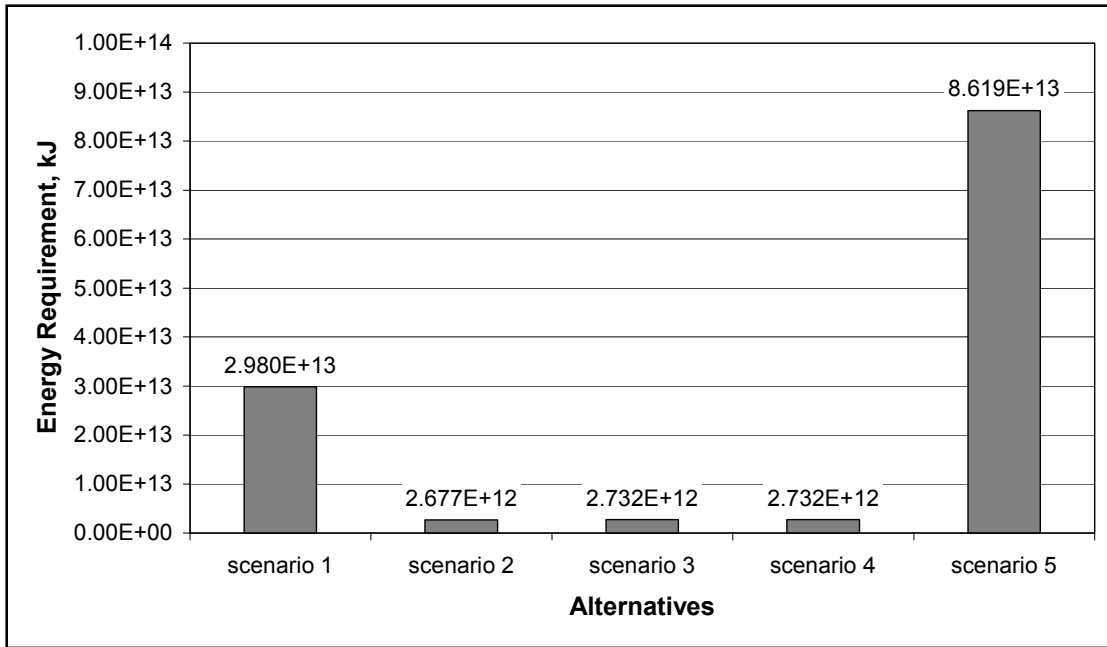


FIGURE 4.10 Energy requirements of each scenario.

The energy requirement of each alternative has been depicted on **Figure 4.10**. The first and the fifth alternatives require considerable amount of energy. This can be attributed to huge energy requirement associated with transportation of produced water to processing facilities, as these are the only scenarios in which no water separation has been considered.

The energy requirement for each alternative has been used to calculate the operating expenditure, OPEX, for each alternative in the economical analysis.

The capital expenditure associated with construction of pipeline system for each alternative has been shown on **Figure 4.11**. The required expenditure for construction of the topside facilities has not been considered since the field effluents will be processed in already existing facilities. The fourth alternative has the maximum CAPEX because of double transfer line and riser system. The third and the fifth alternatives have next highest CAPEX because of the large pipeline required to transfer the fluids over a long distance.

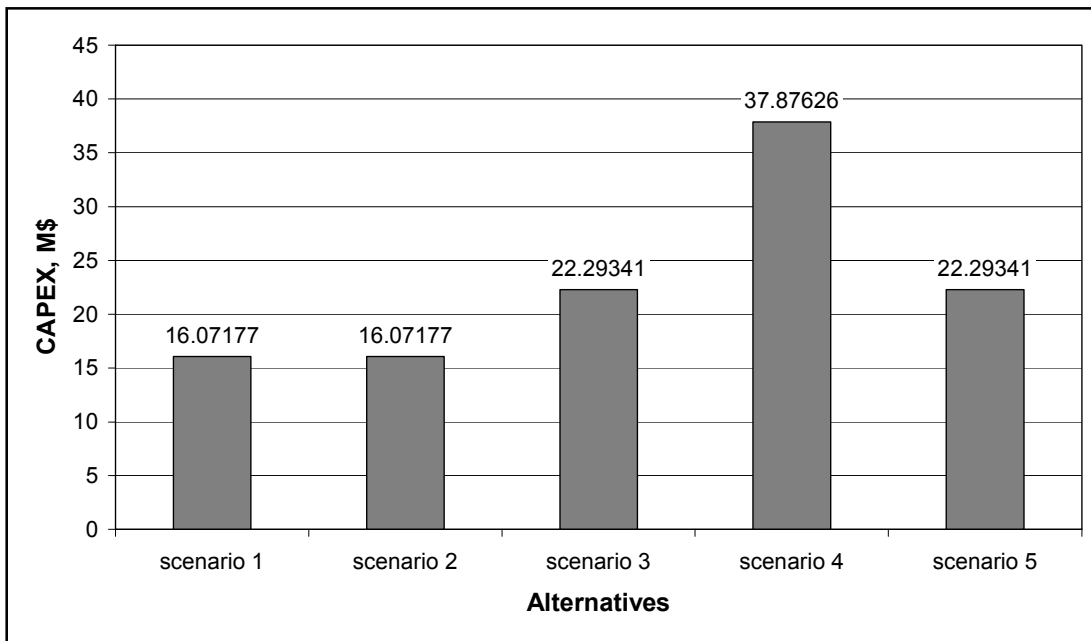


FIGURE 4.11 CAPEX associated to pipeline construction.

Total oil produced by each alternative is depicted on **Figure 4.12**. Second scenario has the highest production. This can be attributed to both longer production period and more favorable surface separation. The later is however the more important factor as production is not significant at final years.

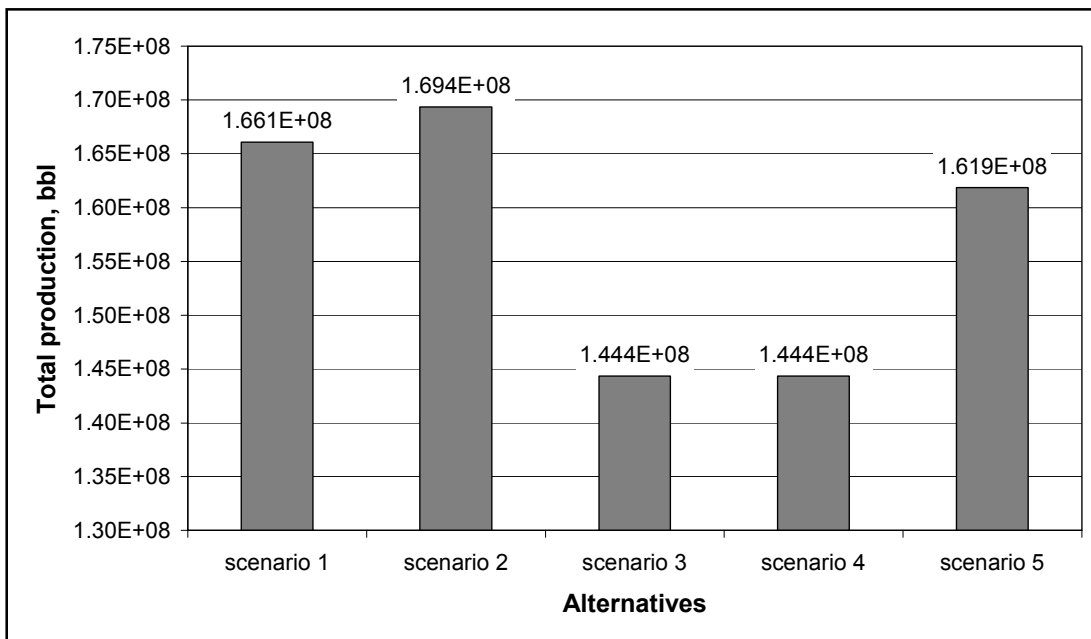


FIGURE 4.12 Cumulative oil production for each alternative.

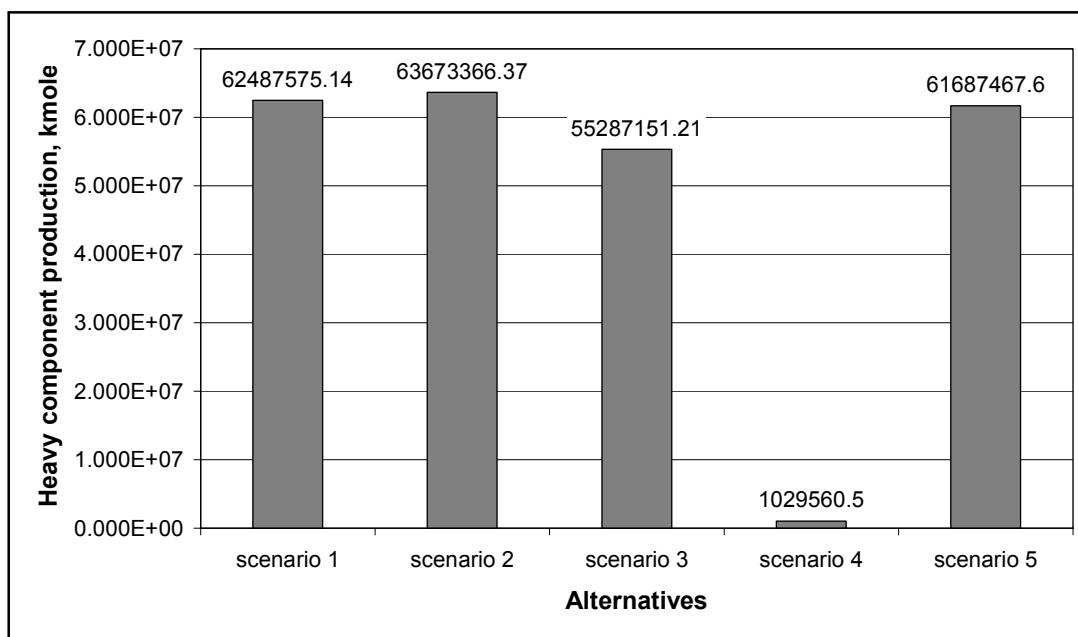


FIGURE 4.13 Total heavy component production in each alternative.

The effects of more favorable surface separation can be verified by looking at total heavy component production from each alternative. As shown on **Figure 4.13**, the second scenario produces the maximum total moles of heavy components. The molar density of the produced liquid hydrocarbon in the second scenario is minimum among all alternatives, causing the maximum volumetric production of liquid hydrocarbons.

The cumulative gas production in each alternative is the last parameter required before an economical analysis can be performed. **Figure 4.14** shows the total gas production expected from each scenario. The first alternative produces more gas than all other alternatives combined. The reason is related to the pressure at which the produced water is separated from hydrocarbons. In alternatives 2 to 4, water is separated at wellhead pressure, while in first and fifth alternatives water is separated at much lower pressures. Because of higher solubility of light hydrocarbons in water at high pressures, the water that is separated at the seabed will contain much more light hydrocarbons than the water separated at lower pressures. Consequently, considerable amount of light hydrocarbons are wasted in alternatives 2 to 4. Although water is separated at lower pressures in the fifth alternative, the gas production expected from this scenario is not as big as the first alternative because of less favorable separation condition at shore that results in higher molar density of the produced gas.

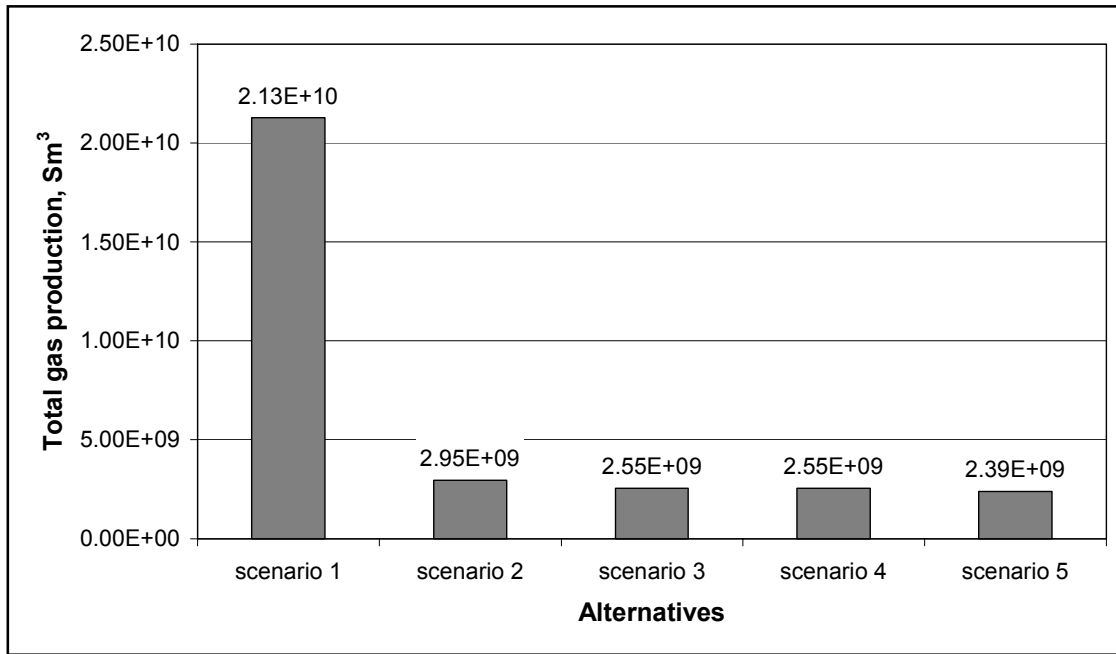


FIGURE 4.14 Cumulative gas production for each alternative.

The final comparison of the alternatives is carried out by calculating the net present value of each alternative. The net present value is calculated using the following relation,

$$NPV = \sum_{i=1}^n \frac{\left(\frac{\text{Oil}}{\text{Price}} \right)(i) \times \left(\frac{\text{Oil}}{\text{Prod}} \right)(i) + \left(\frac{\text{Gas}}{\text{Price}} \right)(i) \times \left(\frac{\text{Gas}}{\text{Prod}} \right)(i) - \text{CAPEX}(i) - \text{OPEX}(i)}{(1+r)^n}$$

where r is the interest rate.

The net present value calculated for the five alternatives has been depicted on **Figure 4.15**. As can be seen from this figure, the second alternative has the highest NPV and is therefore the best alternative to produce the field under study. Its high oil production rate and its low operating costs make it a better alternative than the first scenario. **Table 4.7** shows the parameters used in NPV calculations.

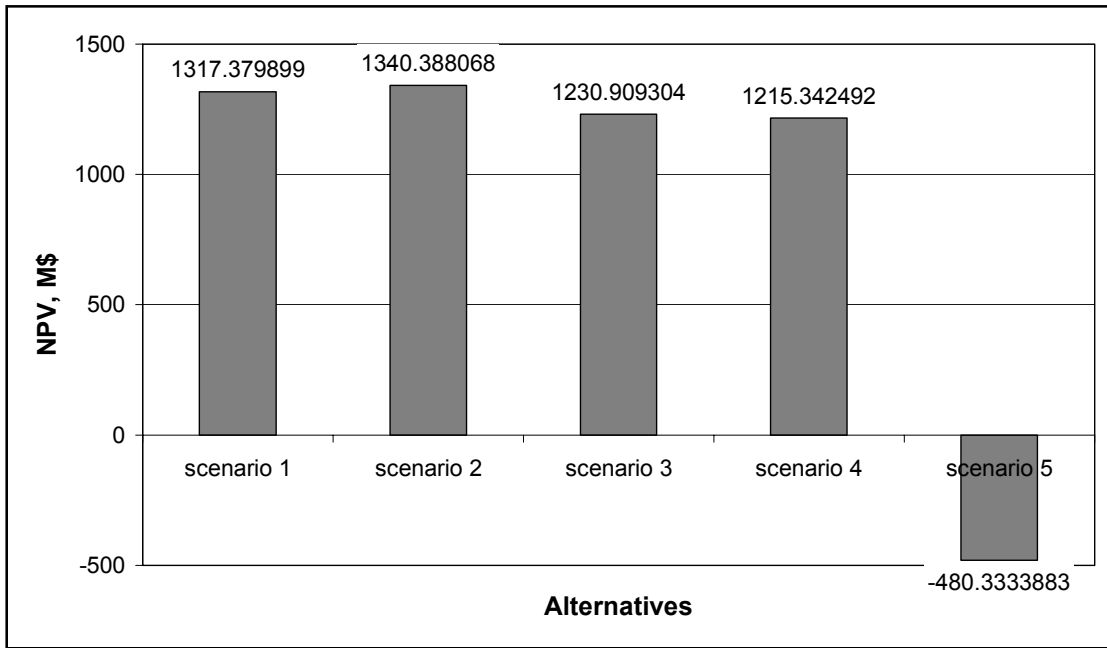


FIGURE 4.15 Net present value of each scenario.

TABLE 4.7 Parameters used in calculation of net present values.

Interest rate, r	10 %
Oil price, \$/bbl	14.00
Gas price, \$/Sm ³	0.061
Piping cost, \$/in m	9.587
Energy cost, \$/kWh	0.149

4.6 Determination of flow regimes

Depending on the number of present phases and their superficial velocities, different production scenarios will lead to different flow regime situations in the production system. In some situations, undesirable flow regimes such as slug flow can predominate large parts of the system causing problems such as erosion or severe control problems in the system. These problems can cost so much to avoid that they may influence the final decision on the best alternative. A compositional flow regime determination routine called DoFazi was developed and integrated with the rest of the system in order to study existing flow regimes during the project life for all alternatives. The results of this study have been summarized in **Figure 4.16**.

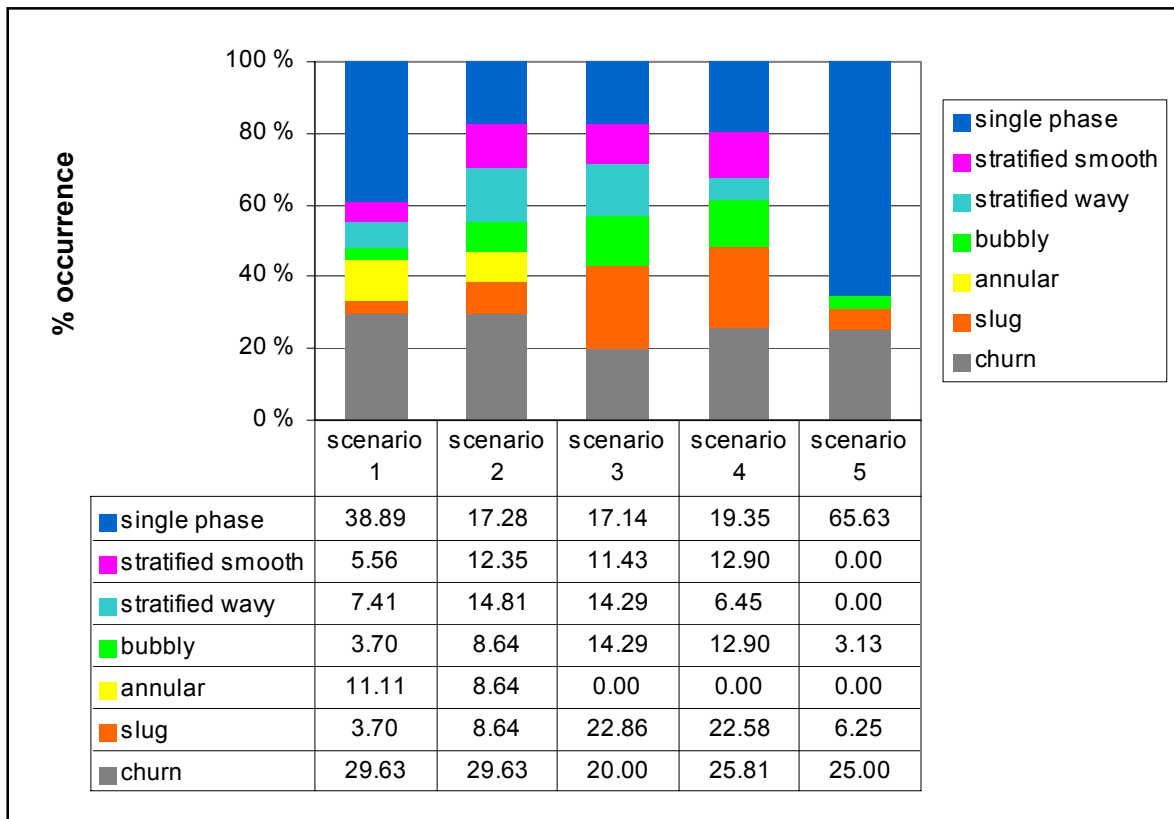


FIGURE 4.16 Flow regime occurrence in each alternative.

Single phase flow is the dominant flow regime in the first and the last scenarios because of presence of large amounts of water in the system shortly after the production begins. The flow regimes that occur at high velocities constitute a relatively smaller percent in the fifth scenario than the first, mainly since larger pipe sizes are used in the fifth scenario.

Scenarios 3 and 4 produce very similar flow regimes because of the fact that these two alternatives are basically the same from the viewpoint of water separation and long transfer line to the shore. Scenario 4, however demonstrates more churn and slug flow situations since after 180 days the whole reservoir effluent is transported in the liquid pipeline alone and this causes high flow velocities.

The first and the second scenarios exhibit more variety of flow regimes than the rest of the alternatives because of the special flow situation brought to the system by the downward flow movement in the gas transfer line. Because of smaller pipe sizes used in scenario 2, it exhibits more undesirable flow situations than the first scenario.

Table 4.8 shows the final ranking of the alternatives from the viewpoint of the dominant flow regime. Although the second alternative has the third rank in exhibiting fewer undesirable flow regimes, i.e., slug and churn flow, the difference is not so huge to disqualify this scenario as the best alternative.

TABLE 4.8 Ranking alternatives according to occurrence of undesirable flow regimes.

Scenario	Occurrence of undesirable flow regime, %	Rank
1	33.33	2
2	38.28	3
3	42.86	4
4	48.39	5
5	31.25	1

4.7 Remarks

There are a number of conclusions that can be made based on the results obtained from the analysis presented in this section:

1. Water separation at the seabed is recommended for fields with high water-cuts. The benefits associated to water separation increase as the water depth increases. Separation at the bottomhole can be even more promising for high water-cut fields, since excessive pressure drop occurs in vertical parts of the flow system. This can be more critical in low pressure reservoirs.
2. The final decision on the best alternative is directly related to oil and gas prices and costs associated to energy and construction of required piping system. A proper estimation of the piping cost is considered the most critical, as prices of oil and gas, and energy costs can be estimated more accurately.

4.8 General observations regarding production of deepwater prospects

The discussion of the previous section mainly focused on ranking the five proposed scenarios and selecting the best one. It is however possible to generalize the results obtained from case study 5 to make some observations regarding the production from deepwater prospects.

Production from deepwater fields involves exposure of the pipelines to cold water over longer distances. Therefore, temperature drop and problems associated with it are considered as most important factors that differentiate deepwater from shallow water production. **Figure 4.17** depicts the expected temperature profile along the production system for a typical well in the field. The temperature profile is divided into three segments. The first segment represents the temperature profile in the vertical part of the well. The well is 1569 meter long. The second segment belongs to the 50 km long transfer line that connects the reservoir to the main production manifold. The third segment represents the temperature profile in the 1500 m long riser. As can be seen from this figure, although the transfer line is more than 30 times longer than either the well or the riser, still the temperature drop along the transfer line is nearly as great as that of either the well or the riser. Considering the fact that the thermal conductivity and thickness of the insulations that have been designed for all these three segments are basically the same, higher heat transfer rate in the wellbore and the riser should be attributed to the flow situations that exist in these vertical pipe segments. While low velocity regimes such as stratified flow exist in the transfer line, the dominant flow regimes in the well or the riser are usually high velocity regimes such as churn or slug flow. Convective heat transfer coefficient of the fluid flowing in the pipe is believed to increase considerably at elevated velocities and can increase the rate of heat transfer to several orders of magnitude (Hasan, 1991).

It can be concluded that the critical part of the system in which a considerable temperature loss can occur over a relatively short distance is the pipe segment in which a vertical or near vertical flow exists. The horizontal parts of the system are considered less critical from the viewpoint of temperature drop.

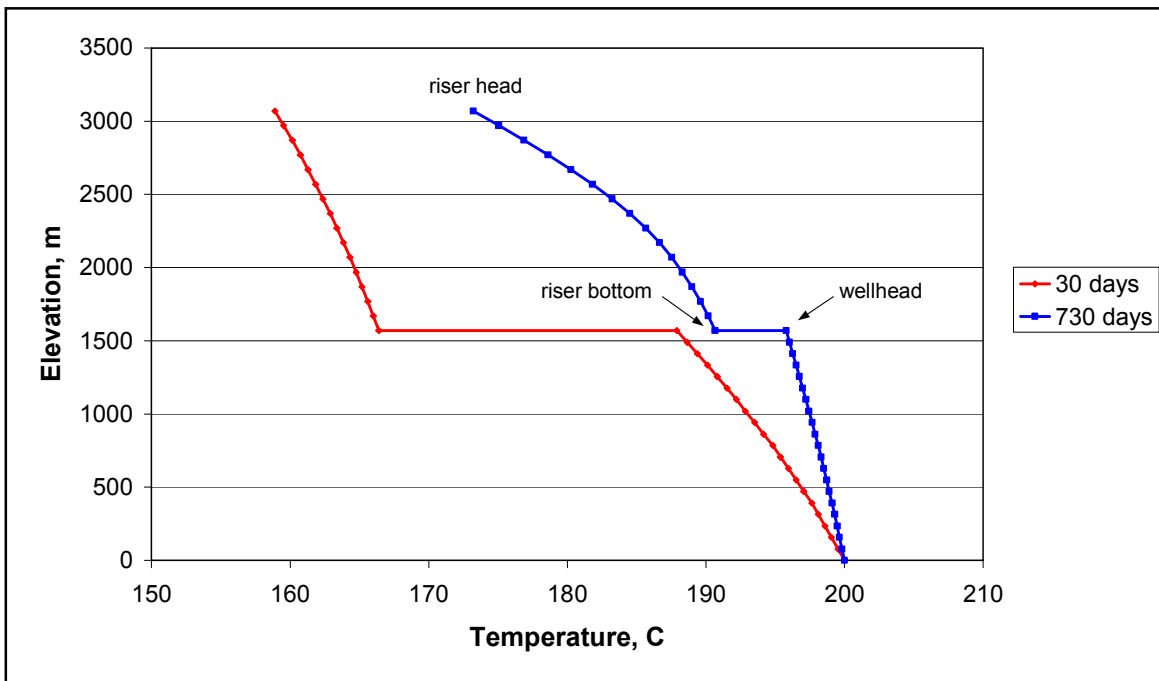


FIGURE 4.17 Temperature profile along the production system.

Consequently, in design of the longer risers that are utilized in production of the deepwater fields, one should consider the application of more effective insulators. It is equally important to size the riser in a way to prevent formation of undesirable flow regimes that cause increased temperature losses.

The considerable difference that is observed between the temperature profiles at 30 and 730 days is basically because of the change in the thermal properties of the flowing fluids caused by increase in water production. Water has a specific heat that is nearly five times smaller than the specific heat of liquid hydrocarbons. As the water-cut increases, the average specific heat of the flowing liquid decreases causing a decrease in convective heat transfer coefficient inside the pipe. This will in turn cause the total heat transfer rate across the pipe wall to decrease. The result is a decrease in temperature loss to the environment as depicted on **Figure 4.17**. Therefore, a reduction in temperature drop can be expected as a result of increase in water-cut in the system.

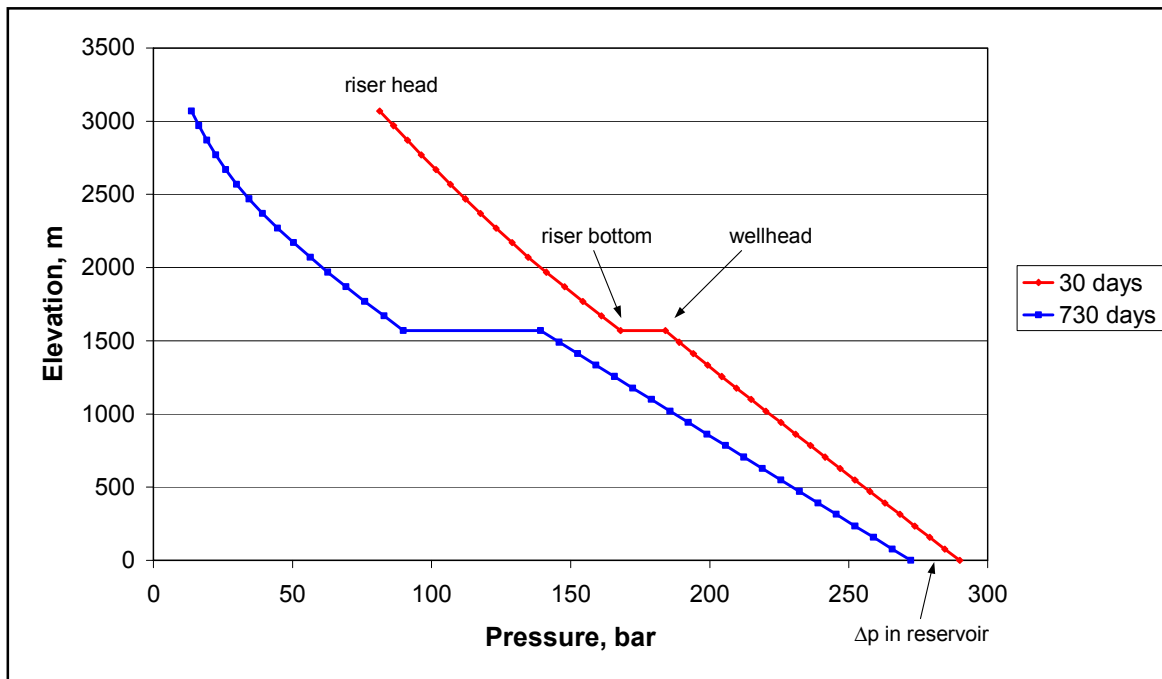


FIGURE 4.18 Pressure profile along the production system.

Analysis of pressure profiles across the system during the life of the project is essential in designing the required piping network. The pressure profile along the production system for a typical well is shown in **Figure 4.18**. As expected, the pressure drop along the transfer line is much smaller compared to pressure drop along the well or the riser.

As the water-cut increases, the average density of the producing fluid increases and a higher pressure drop is observed in vertical parts of the system. As can be seen from this figure, the wellhead pressure at 730 days is smaller than pressure at the bottom of the riser at 30 days. Therefore, the network cannot converge to the delivery pressure, i.e., the riser bottom pressure, which was set at 30 days. Consequently, a lower pressure should be set at the riser bottom at 730 days. The increase in pressure drop across the transfer line at 730 days is because of the new pressure setting at the network delivery point.

Chapter 5

Conclusions

5.1 Observations

The main objective of this study was to demonstrate that an integrated simulation system can be developed using the available software technology and that integrated simulation is a better approach to petroleum engineering simulation. To achieve this, a platform was recommended and a hybrid application was accordingly developed and put into test. The hybrid application or the field simulator was shown to be capable of handling a relatively extended range of problems encountered in simulation of petroleum production systems.

It was recognized that the integrated system should acquire new types of specifications and meet new demands in order to be viable. An efficient and practical integrated simulation system should essentially possess the following specifications:

1. It should be easy to develop, use, maintain, and upgrade.
2. Simulation models for major system components should either exist or can be integrated readily to the system in order to generate a comprehensive modeling tool.
3. The integrated simulator should not be highly CPU intensive.
4. It should contain or readily accommodate techniques that reduce simulation time in all integrated softwares.
5. Auxiliary programs that are integrated to the system for special purposes should not restrict system functionalities or increase CPU requirements to a level that an integrated solution becomes infeasible. This is a critical requirement in components that involve iterative procedures.

The proposed platform possesses most of the mentioned specifications because of the employed integration method and offers several possibilities to include other required features due to its open software structure.

5.2 Overall conclusions

Based on the experiences gained through development and application of the integrated system and simulation results presented in this thesis, the following overall conclusions can be made:

1. The petroleum production system is an integrated system and should preferably be simulated using an integrated model. An integrated simulation model is a faster, and more accurate tool in simulating such integrated systems.
2. With the available software technology, it is possible to create integrated models of petroleum production systems by coupling together available commercial softwares to model system components. Solution of the integrated system is not in general more difficult than solution of individual components, since each subsystem performs its own calculation.
3. Absence of simplifying assumptions in such integrated systems makes the results more realistic and general.
4. An integrated system developed based on the proposed platform, offers considerable flexibility in modeling the production system and its open structure facilitates integration of other software components to carry out special studies.
5. Satisfactory performance was obtained from application of genetic algorithms in the developed integrated system. VisualGene was shown to be a good example of a search routine that utilizes the power of genetic search to a great extent. Its use considerably widened functionalities of the integrated simulator.
6. The proposed network simulation method simplified development and utilization of the integrated system by avoiding integration of a third simulator. The method has satisfactory performance and is easy to construct.

7. In an integrated software, the difference in calculation results obtained from participating simulators is not overlooked and wrong interpretation of the results is avoided when such system is used. It is in general possible to find methods to eliminate or reduce calculation inconsistency in integrated systems.
8. Solution to different classes of problems is usually achieved by minor modifications of existing models or by integration of new components to the integrated system.
9. The automated nature of performing calculations and data transfer in dynamically linked systems makes it possible to carry out studies that involve repeated calculation of one or several subsystems. It is practically impossible to perform such studies using traditional approaches that involve manual data transfer or approaches in which data is read from output files.
10. The modular characteristic of an integrated system facilitates considerably the job of maintenance, modification, and upgrading of the system, provided that such changes are performed in a way that communication protocols are not violated.

5.3 Recommendations for further work

Facilities that the developed hybrid application offers in modeling and simulation of petroleum production systems probably generate the incentive to focus the future activities on the proposed platform while at the same time it is basically possible to develop an integrated system in numerous other ways.

The inherent flexibility of the proposed platform allows modification of the developed prototype to overcome its shortcomings. Based on the experiences obtained from using the prototype, there are two basic modifications that can be proposed to further enhance the capabilities of the developed tool.

The first modification is related to HYSYS valve model that has been used to simulate chokes in the gathering network simulation case. The HYSYS steady state valve model is simply an element to create a given pressure drop in a material stream without changing the stream's flow rate. In reality, flow

through an opening is related to the pressure drop across it. As a useful modification, it is recommended to develop a choke model in HYSYS as a User-Defined unit operation to substitute the HYSYS valve model (HYSYS documentation, 2000). The Sachdeva choke model is recommended since it develops a single equation for both critical and sub-critical flow through the choke without pressure jump at critical point (Sachdeva, 1986). Using this model, flow rate through the choke can be determined as a function of backpressure exerted by the choke on the wells. Calculated stream flow rate can then be transferred to the reservoir simulator to modify the well production rate.

The second modification is related to fluid characterizations that are used in reservoir and process simulators. As was mentioned in Chapter 2, while 8 to 12 components are usually quite enough to simulate the reservoir behavior, 15 to 25 components are usually used in process simulation. To speed up the reservoir calculations in one hand and increase the precision of process calculations on the other hand, it is recommended to integrate a software into the system to optimize the characterization in the integrated system. The method proposed by Hoda provides an excellent solution to the mentioned problem (Hoda, 2002).

Nomenclature

a	EOS parameter defined in Equation 2.20
A	pipe cross-section area, m^2
b	EOS parameter defined in Equation 2.21
B_g	gas formation volume factor, <i>reservoir</i> $m^3 / Std. m^3$
B_o	oil formation volume factor, <i>reservoir</i> $m^3 / Std. m^3$
C_L	constant in Equation [2], Appendix C
d	bubble diameter, m
d	distance between individuals in Equation 3.1
D	tube inside diameter, m
d_C	bubble diameter on transition boundary, m
d_{CB}	lower limit, critical bubble size, m
d_{CD}	upper limit, critical bubble size, m
f	friction factor
f	fitness in Equation 3.1
F	Froude number
g	gravitational acceleration, m / s^2
h_L	equilibrium liquid level, m
p	pressure, kPa
R	liquid holdup, universal gas constant
R_s	liquid holdup in liquid slug
R_s	solution gas-oil ratio in black oil model, $Std. m^3 / Std. m^3$
R_{sm}	minimum liquid holdup in liquid slug
s	sheltering coefficient in Equation [20], Appendix C
s	share in Equation 3.1
T	absolute temperature
u	constant used in HYSYS EOS formulation
U	velocity, m / s
U_0	bubble rise velocity, m / s
v	molar volume, $m^3 / kmole$
V	volume, m^3
w	constant used in HYSYS EOS formulation
x	direction along the pipe
x	string or individual in Equation 3.1
X	Martinelli parameter defined in Equation [12a], Appendix C
Y	Martinelli parameter defined in Equation [12b], Appendix C
Z	gas compressibility factor

Greek

α	gas void fraction
α_s	gas void fraction within liquid slug
β	pipe deviation angle from horizontal
ε	tube absolute roughness, m
γ	distortion coefficient of the bubble
κ	parameter used in HYSYS EOS formulation, Equation 2.14
ν	kinematic viscosity, m^2 / s
ν	chromosome
ρ	density, kg / m^3
σ	surface tension, kg / s^2
ω	acentric factor
Ω_a	EOS constant, Equation 2.20
Ω_b	EOS constant, Equation 2.21
ξ	EOS parameter, Equation 2.10

Subscripts and superscripts

c	critical
G	gas
GS	gas superficial
i	component i
L	liquid
LS	liquid superficial
M	mixture
R	reduced
s	shared
\sim	dimensionless, translated parameter

References

1. Aarts, E., and Korst, J.: *Simulated Annealing and Boltzman Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons, New York, 1989.
2. Ackley, D.: *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, Boston, 1987.
3. Asheim, H.: "Numerical Optimization of Production Strategies for Slightly Compressible Petroleum Reservoir", *SINTEF report no. STF28 A81001*, NTH, Trondheim, Norway, 1980.
4. Aziz, K., Govier, G. W., and Fogarasi, M.: "Pressure Drop in Wells Producing Oil and Gas," *J. Cdn. Pet. Tech.*, pp. 38-48, July-Sept., 1972.
5. Bäck, T.: *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.
6. *Baker Jardine press release*, <http://www.bakerjardine.com/info/news007.html>
7. Barnea, D.: "A Unified Model for Predicting Flow-Pattern Transitions for the Whole Range of Pipe Inclinations," *Int. J. Multiphase Flow*, **13**, 1-12, 1987.
8. Bayer, T.J., Edwards, G.M., and Aziz, K.: "Preconditioned Newton methods for Fully Coupled Reservoir and Surface Facility Models", *paper SPE 49001*, presented at the 1998 SPE Annual Technical Conference, New Orleans, Louisiana, September 27-30, 1998.
9. Bayer, T.J., Edwards, G.M., and Aziz, K.: "A Preconditioned Adaptive Implicit Method for Reservoirs with Surface Facilities", *paper SPE 51895*, presented at the 1999 SPE Reservoir Simulation Symposium, Houston, Texas, February 14-17, 1999.
10. Beggs, H.D., and Brill, J.P. : "A Study of Two-Phase Flow in Inclined Pipes", *J. Petrol. Tech.*, pp. 607, May, 1973.

11. Beliakova, N., van Berkel, J.T., Kulawski, G.J., Schulte A.M., and Weisenborn, A.J.: "Hydrocarbon Field Planning Tool for Medium to Long Term Production Forecasting from Oil and Gas Fields Using Integrated Subsurface – Surface Models", *paper SPE 65160, presented at the SPE European Petroleum Conference, Paris, France, October 24-25, 2000.*
12. Bender, E. : "Simulation of Dynamic Gas Flow in Networks Including Control Loops", *Comp. & Chem. Eng.*, vol. 3, no. 1-4, pp. 611-613, 1979.
13. Bertrand, A., and Horne, R.N.: "Optimal Reservoir Production Scheduling by Using Reservoir Simulation", *SPE Journal*, October, 1983.
14. Biegler, L.T., Grossmann, I.E., and Westerberg, A.W.: *Systematic Methods of Chemical Process Design*, Prentice Hall, 1997.
15. Bittencourt, A.C.: "Reservoir Development and Design Optimization", *paper SPE 38895, presented at the SPE Annual Technical Conference and Exhibition, San Antonio, October 5-8, 1997.*
16. Bohannon, J.M.: "A Linear Programming Model for Optimum Development of Multireservoir Pipeline Systems", *J. Pet. Tech.*, pp. 1429-1436, November, 1970.
17. Bolc, L., and Cytowski, J.: *Search Methods for Artificial Intelligence*, Academic Press, London, 1992.
18. Braunschweig, B., Pantelides, C.C., Britt, H.I., and Sama, S.: "Open Software Architectures for Process Modeling: Current Status and Future Perspectives", *FOCAP99*, July, 1999.
19. Breaux, E.J., Monroe, S.A., Blank, L.S., Yarberry Jr., D.W., and Al-Umran, S.A.: "Application of a Reservoir Simulator Interfaced with a Surface Facility Network: A Case History", *paper SPE 11479, presented at the SPE Middle East Oil Technical Conference, Manama, March 14-17, 1984.*

20. Carroll, J.A., and Horne, R.N.: "Multivariate Optimization of Production Systems", *JPT*, pp. 782-789, July, 1992.
21. Chow, C.V., and Arnondin, M.C.: "Managing Risks Using Integrated Production Models: Process Description", *JPT*, March, 2000.
22. Chong, H., and Coon, J.E.: "An Internally Consistent Approach for Determining the Properties of Lumped Components Using a Cubic Equation of State", *Fluid Phase Equilibria*, **117**, pp. 233-240, 1996.
23. Danesh, A., Donghai, X., and Todd, A.C.: "A Grouping Method to Optimize Oil Description for Compositional Simulation of Gas Injection Processes", *paper SPE 20745, presented at the 65th SPE Annual Technical Conference and Exhibition*, New Orleans, LA, September 23-26, 1990.
24. David, B.T.: "Multi-Expert Systems for CAD", *Intelligent CAD Systems I: Theoretical and Methodological Aspects*, pp. 57-67, Springer-Verlag, Noordwijkerhout, The Netherlands, 1987.
25. De Jong, K.A.: "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Doctoral Dissertation, University of Michigan, *Dissertation Abstract International*, 36 (10), 5140B., University Microfilms No. 76-9381, 1975.
26. Deutman, R., and Van Rijen, M.A.: "Case Study of Integrated Gas Field System Modeling in the North Sea Environment", *paper SPE 38556, presented at the SPE Offshore European Conference*, Aberdeen, Scotland, September 9-12, 1997.
27. Dupuis, P., Robert, J.L., and Ouellet, Y.: "A Modified Element Method for Pipe Network Analysis", *J. Hydraul. Res.*, vol. **25**, no. 1, pp. 27-40, 1987.
28. *Eclipse Reference Manual*, Schlumberger GeoQuest, (2002).
29. Epperly, T.G.W.: *Global Optimization of Nonconvex Nonlinear Programs using Parallel Branch and Bound*, PhD thesis, University of Wisconsin, Madison, 1995.

30. Faissat B., and Duzan, M.C.: “Fluid Modeling Consistency in Reservoir and Process Simulations”, *paper SPE 36932, presented at the 1996 SPE European Petroleum Conference*, Milan, Italy, October 22-24, 1996.
31. Floudas, C.A.: *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*, Topics in Chemical Engineering, Oxford University Press, 1995.
32. *FORGAS User Manual*, Neotechnology Consultants Ltd., (1997).
33. Foster, B.L.: “Maximization in the Oil Industry – A Survey”, *Management Technology*, 4 (1), pp. 26-46, 1964.
34. Frair, L., and Devine, M.: “Economic Optimization of Offshore Petroleum Development”, *Management Sciences*, 21 (12), pp. 1370-1379, 1975.
35. Fujii, H., and Horne, R.N.: “Multivariate Optimization of Networked Production Systems”, *paper SPE 27617, presented at the 1994 SPE Production Operations and Exhibition*, Aberdeen, March 15-17, 1994.
36. Garvin, W.W., Crandall, H.W., John, J.B., and Spellman, R.A.: “Applications of Linear Programming in the Oil Industry”, *Management Science*, **3**, pp. 407-430, 1957.
37. Gayton, P.W., Miller, S.D., and Napalowski, R.: “Innovative Development Engineering Techniques”, *paper SPE 65202, presented at the SPE European Petroleum Conference*, Paris, France, October 24-25, 2000.
38. Gen, M., and Cheng, R.: *Genetic Algorithms and Engineering Design*, John Wiley, New York, 1997.
39. *Genetic Algorithm (GA)*, <http://cuaerospace.com/carroll/ga.html>
40. Geoffrion, A.M.: “Generalized Benders Decomposition”, *Journal of Optimization Theory and Applications*, 10 (4), 237, 1972.

41. Glover, F., and Greenberg, H.: “New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence”, *European Journal of Operational Research*, vol. 39, pp. 119-130, 1989.
42. Goldberg, D.E., and Richardson, J.: “Genetic Algorithms with Sharing for Multimodal Function Optimization: Genetic Algorithms and Their Applications”, *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41-49, Cambridge, MA, 1987.
43. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, Massachusetts, 1989.
44. Gould, T. L.: “Compositional Two-Phase Flow in Pipelines,” *J. Pet. Tech.*, pp. 373-384, March, 1979.
45. Gray, P., Hart, W., Painton L., Philips, C., Trahan, M., and Wagner J.: *A Survey of Global Optimization Methods*. 1997.
<http://www.cs.sandia.gov/opt/survey/>
46. Grefenstette, J., and Baker, J.: “How Genetic Algorithms Work: A Critical Look at Implicit Parallelism”, *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 20-27, Morgan Kaufmann Publishers, San Mateo, CA, 1989.
47. Gregory, G.A., Mandhane, J. and Aziz, K.: “Some Design Considerations for Two-Phase Flow in Pipes”, *J. Can. Petrol. Techn.*, January – March, 1975.
48. Haghghi, K., Mohtar, R., Bralts, V.F., and Segerlind, L.J.: “Linear Formulation for Pipe Network Components”, *Comput. Electron. Agric.*, vol. 7, no. 4, pp. 301-321, 1992.
49. Hallefjord, Å., Asheim, H., and Haugland, D.: “Petroleum Field Optimization: Comments on Models and Solution Techniques”, *Report No. 862331-1*, CHR. Michelsen Institute, December, 1986.
50. Hasan, A.R., and Kabir, C.S.: “Heat Transfer During Two-Phase Flow in Wellbores – Parts I & II”, *papers SPE 22866 & 22948, presented at the SPE Annual Technical Conference and Exhibition*, Dallas, October 6-9, 1991.

51. Haugen, E.D., Holmes, J.A., and Selvig, A.: "Simulation of Independent Reservoirs Coupled by Global Production and Injection Constraints", *paper SPE 29106, presented at the 13th SPE Symposium on Reservoir Simulation*, San Antonio, Texas, February 12-15, 1995.
52. Hepguler, F., Barua, S., and Bard, W.: "Integration of a Field Surface and Production Network With a Reservoir Simulator," *SPE Computer Applications*, **88**, June, 1997.
53. Hoda, M.F., *The engineering of petroleum streams*, PhD Thesis, Norwegian University of Science and Technology, 2002.
54. Holland, J.H.: *Adaptation in natural and artificial Systems*, Ann Arbor, The University of Michigan Press, 1975.
55. Holland, J.H.: "Genetic Algorithms", *Scientific American*, **66**, July 1992.
56. Homaifar, A., Qi, C., and Lai, S.: "Constrained Optimization via Genetic Algorithms", *Simulation*, vol. 62, no. 4, pp. 242-254, 1994.
57. Hooi, H.R., Goobie, L., Cook, R., and Choi, J.: "The Integrated Team Approach to The Optimization of a Mature Gas Field", *paper SPE 26144, presented at the SPE Gas Technology Symposium*, Calgary, Canada, June 28-30, 1993
58. Horst, R., Pardolas, P.M., and Thoai, N.V.: *Introduction to Global Optimization*, vol. 3 of *Nonconvex Optimization and its Application*, Kluwer, 1995.
59. Huseby, A.B., and Brækken, E.: "An Integrated Risk Model for Reservoir and Operational Costs with Applications to Total Value Chain Optimization", *paper SPE 65145, presented at the SPE European Petroleum Conference*, Paris, France, 24-25, October, 2000.
60. *HYSYS.Process V.2.2.2 Documentation*, Hyprotech, (2000).
61. Ingber, L.: "Simulated Annealing: Practice versus Theory", *Mathematical Computer Modeling*, 18 (11), pp. 29-57, 1993.

62. Isaacs, L.T., and Mills, K.G. : “Linear Theory Methods for Pipe Network Analysis”, *J. Hydraul. Div. ASCE*, vol. **106**, no. HY7, pp. 1191-1201, 1980.
63. Johansen, O., Leporcher, E., and Lezeau, P.: “Benefits of Adding Energy to the Wellstream at the Mudline”, *paper OTC 12012, presented at the 2000 Offshore Technology Conference*, Houston, Texas, May 1-4, 2000.
64. Joines, J., and Houck, C.: “On the Use of Non-stationary Functions to Solve Nonlinear Constrained Optimization Problems with GAs”, *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp. 579-584, IEEE Press, Orlando, FL, 1994.
65. Kim, J.H., and Myung, H.: “A Two-Phase Evolutionary Programming for General Constrained Optimization Problem”, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, San Diego, 1996.
66. Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M.P.: “Optimization by Simulated Annealing“, *Science*, **220**, 671, 1983.
67. Laarhoven, P.J.M., and Aarts, E.H.L.: *Simulated Annealing: Theory and Applications*, Kluwer, 1987.
68. Lamey, M.F., Schoppa, W., Stingl, K.H., and Turley, A.J.: “Dynamic Simulation of the Europa and Mars Expansion Projects: A New Approach to Coupled Subsea and Topsides Modeling”, *paper SPE 56704, presented at the 1999 SPE Annual Technical Conference and Exhibition*, Houston, Texas, October 3-6, 1999.
69. Leibovici, C.F., Barker, J.W., and Waché, D.: “Method for Delumping the Results of Compositional Reservoir Simulation”, *SPE Journal*, 5 (2), June, 2000.
70. Liepins, G., Hilliard, M., Richardson, J., and Pallmer, M.: “Genetic Algorithm Application to Set Covering and Traveling Salesman Problems”, Brown, Editor, OR/AI, *The Integration of Problem Solving Strategies*, 1990.

71. Linthorst, S.J.M., Coutts, D., and Van Stiphout, M.T.: "The Brent Full Field Model: The Reservoir Management Tool for Depressurization", *paper SPE 38474, presented at the SPE Offshore European Conference, Aberdeen, Scotland, September 9-12, 1997.*
72. Metropolis, N.: "Equation-of-State Calculations by Fast Computing Machines", *J. Chem. Phys.*, **21**, 1087, 1953.
73. Michalewicz, Z.: *Genetic Algorithm + Data Structure = Evolution Programs*, 2nd ed., Springer-Verlag, New York, 1994.
74. Michalewicz, Z.: "Genetic Algorithms, Numerical Optimization, and Constraints", *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco, 1995.
75. Michalewicz, Z., Dasgupta, D., and Le Riche, R.G.: "Evolutionary Algorithms for Industrial Engineering Problems", *Intern. J. Comp. Ind. Eng.*, vol. 3, no. 4, 1996.
76. Mogensen, A.C., and Trick, M.D.: "Application of Integrated Reservoir Simulation and Pipeline Network Modeling to the Sexsmith Gas-Condensate Field," *paper SPE 40002, presented at the 1998 SPE Gas Technology Symposium, Calgary, Alberta, Canada, March 15-18, 1998.*
77. Myung, H., and Kim, J.H.: "Hybrid Evolutionary Programming for Heavily Constrained Problems", *Networks*, vol. 26, pp. 231-241, 1995.
78. Nazarian B., and Golan M.: "Recipes for Integrating a Reservoir Model into a Process Modeling Environment", *presented at the 2nd International Oil, Gas, and Petrochemicals Congress, Tehran, Iran, May 15-17, 2000.*
79. Nazarian B.: *Proposing a Tool for Simulation of Production Piping Network*, Report on Network Solution of Production Pipe System self-study course, IPT, Norwegian University of Science and Technology, Trondheim, Norway, August, 2000.
80. Nesvold, R.L., Herring, T.R., and Currie J.C.: "Field Development Optimization using Linear Programming Coupled with Reservoir

- Simulation – Ekofisk Field”, *paper SPE 36874, presented at the 1996 SPE European Petroleum Conference, Milan, Italy, October 22-24, 1996.*
81. *OLGA User’s Manual*, Scandpower a/s, (2000).
 82. *Open-Eclipse Developers Kit*, Schlumberger GeoQuest, (2002).
 83. Orvosh, D., and Davis, L.: “Using a Genetic Algorithm to Optimize Problems with Feasibility Constraints”, *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE Press, Orlando, FL, 1994.
 84. *OTTIS Documentation*, AspenTech, (2000).
 85. Palke, M.R., and Horne, R.N.: “Nonlinear Optimization of Well Production Considering Gas Lift and Phase Behavior”, *paper SPE 37428, presented at the 1997 SPE Production Operation Symposium, Oklahoma City, Oklahoma, March 9-11, 1997.*
 86. Pan, Y., and Horne, R.N.: “Improved Methods for Multivariate Optimization of Field Development Scheduling and Well Placement Design”, *paper SPE 49055, presented at the 1998 Annual Technical Conference and Exhibition, New Orleans, Louisiana, September 27-30, 1998.*
 87. Pardalos, P.M., and Rosen, J.B.: “*Constrained Global Optimization: Algorithms and Applications*”, No. 268 in LNCS, Springer-Verlag, 1987.
 88. Pieters, J., and Por, J.A.G.: “Total System Modeling, A Tool for Effective Reservoir Management of Multiple Fields with Shared Facilities” *paper SPE 30442, presented at the Offshore European Conference, Aberdeen, September 5-8, 1995.*
 89. *PVM Documentation*, <http://www.epm.ornl.gov/pvm>
 90. Reid, R.C., Prausnitz, J.M., and Poling, B.E.: *The Properties of Gases and Liquids*, 4th ed., McGraw-Hill Book Company, New York, 1987.

91. Rothman, D.H.. “Nonlinear Inversion, Statistical Mechanics and Residual Statics Estimation”, *Geophysics*, **50**, 2784, 1985.
92. Sachdeva, R., Schmidt, Z., Brill, J.P., and Blais, R.M.: “Two-Phase Flow Through Chokes”, paper SPE 15657, *presented at the 61st Annual Technical Conference and Exhibition*, New Orleans, Louisiana, 1986.
93. Schiozer, D.J.: *Simultaneous Simulation of Reservoir and Surface Facilities*, PhD Thesis, Stanford University, 1994.
94. Schulze-Riegert, R.W., Axmann, J.K., Haase, O., and Rian, D.T.: “Evolutionary Algorithms Applied to History Matching of Complex Reservoirs”, *paper SPE 77301, presented at the SPE Reservoir Simulation Symposium*, Houston, February 11-14, 2001.
95. Schwefel, H.: *Evolution and Optimum Seeking*, John Wiley & Sons, New York, 1994.
96. Sen, M.K., Datta-Gupta, A., Stoffa, P.L., Lake, L.W., and Pope, G.A.: “Stochastic Reservoir Modeling Using Simulated Annealing and Genetic Algorithms”, *paper SPE 24754, presented at the 1992 SPE Annual Technical Conference and Exhibition*, Washington, October 4-7, 1992.
97. Sharif, A., MacDonal, A.C., Kjørholt, O., and Gulbrandsen, S.: “Integrated Reservoir Characterisation and Uncertainty Analysis Using Stochastic Methods and Streamline Solutions”, *paper SPE 65178, presented at the SPE European Petroleum Conference*, Paris, France, October 24-25, 2000.
98. Singh, K.: *Modeling Compositionally-Sensitive Reservoirs*, PhD Thesis, Norwegian University of Science and Technology, 2002.
99. Smith, A., and Tate, D.: “Genetic Optimization Using a Penalty Function”, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 499-505, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
100. Smith, E.W.B.: *On the Optimal Design of Continuous Processes*, PhD Thesis, Imperial College of Science, Technology and Medicine, 1996.

101. Taitel, Y. M., Barnea, D., and Dukler, A. E.: "Modeling Flow Pattern Transitions for Steady Upward Gas-Liquid Flow in Vertical Tubes," *AIChE J.*, 345-354, May, 1980.
102. Thornton, A.C.: "Genetic Algorithm versus Simulated Annealing: Satisfaction of Large Sets of Algebraic Mechanical Design Constraints", *Artificial Intelligence in Design*, Kluwer Academic Publishers, the Netherlands, 1994.
103. Trick, M.D.: "A Different Approach to Coupling a Reservoir Simulator with a Surface Facilities Model," *paper SPE 40001, presented at the 1998 SPE Gas Technology Symposium*, Calgary, Alberta, Canada, March 15-18, 1998.
104. Vitanen, S.: *Some New Global Optimization Algorithms*, PhD Thesis, Åbo Akademi University, 1997.
105. Wattenbarger, R.A.: "Maximizing Seasonal Withdrawal from Gas Storage Reservoirs", *J. Pet. Tech.*, pp. 994-998, 1970.
106. Weisenborn, A.J., and Schulte, A.M.: "Compositional Integrated Subsurface-Surface Modeling", *paper SPE 65158, presented at the SPE European Petroleum Conference*, Paris, France, October 24-25, 2000.
107. Whitson, C.H.: *SPE Phase Behavior Monograph*, Trondheim, Norway, 1994.
108. Whitson, C.H.: "PVTx: An Equation of State Based Program for Simulating and Matching PVT Experiments with Multiparameter Nonlinear Regression", Pera a/s, Trondheim, Norway, 1995.
109. Whitson, C.H., personal communications, 2001
110. Whitson, C.H., personal communications, 2002.
111. Wood, D.J., and Rayes, A.G.: "Reliability of Algorithms for Pipe Network Analysis", *J. Hydraul. Div. ASCE*, vol. 107, no. HY10, pp. 1145-1161, 1981.

Appendix A

Automation

Automation techniques used in creation of the hybrid application are explained in this appendix.

Automation is an extensive subject and this appendix covers only those techniques that were used in this study in connection with HYSYS.

1 HSYSY Automation code

Automation, defined in its simplest terms, is the ability to drive one application from another. For example, the developers of Product *A* have decided in their design phase that it would make their product more usable if they exposed Product *A*'s objects, thereby making it accessible to Automation. Since Products *B*, *C* and *D* all have the ability to connect to applications that have exposed objects, each can programmatically interact with Product *A*.

HYSYS exposes a considerable number of its objects. That makes it a very powerful tool in the design of hybrid solutions. Since access to an application through Automation is largely language-independent, anyone who can write code in Visual Basic, C++ or Java can write applications that will interact with HYSYS.

Automation is a broad programming area and offers much functionality, but only those programming techniques that have been used in this study are explained here.

2 Visual Basic Automation syntax

An object in Visual Basic is basically a variable and should be declared. Objects can be declared using the generic type identifier object. The preferred method however uses the type library reference to declare the object variables by an explicit object name. This method has been used in this study. Once a reference to the type library has been established, the actual name of the object as it appears in the type library can be used. This is called early binding. It offers some advantages over late binding and is therefore used here. **Example A.1** clarifies these object declarations.

Example A.1. *Object declaration*

Late Binding:

```
Public hyCase As Object  
Public hyStream As Object
```

Early Binding:

```
Public hyCase As SimulationCase  
Public hyStream As ProcessStream
```

The path that is followed to get to a specific property may involve several objects. The path and structure of objects are referred to as the object hierarchy. In Visual Basic, the properties and methods of an object are accessed by hooking together the appropriate objects through a *dot operator* (.) function. Each *dot operator* in the object hierarchy is a function call. In many cases, it is beneficial to reduce the number of calls by setting intermediate object variables.

Connections or references to object variables are made by using the Set keyword as is shown in **Example A.2**.

Example A.2. *Set keyword*

Assuming hycase is set to the SimulationCase object

```
Dim hyStream As ProcessStream  
Set hyStream = hycase.Flowsheet.MaterialStreams.Item(0)
```

In order to begin communication between the client and server applications, an initial link to the server application must be established. In HYSYS, this is accomplished through the starting objects; Application or SimulationCase.

The CreateObject function will start a new instance of the main application. CreateObject is used in HYSYS with the *HYSYS.Application* class as specified in the type library. This connects to the main application interface of HYSYS.

The GetObject function will open a specific document in the currently running instance of the server application. If the application is not running, then a new instance of the application will be started. If a specific document is not specified with the GetObject function, the current instance of the application is connected or a new instance of the application is started. **Example A.3** explains the use of these two functions.

Example A.3. *CreateObject and GetObject*

For application objects:

```
Set applicationobj = CreateObject("HYSYS.Application")
```

or

```
Set applicationobj = GetObject( , "PROGRAM.Application")
```

For document objects:

```
Set documentobj=GetObject("c:\filepath", "PROGRAM.Document")
```

3 Starting a HYSYS case through Automation

In **Example A.4**, `hyCase` is declared as type object so it will be using late binding. The `hyCase` variable is connected to a HYSYS case by using the `GetObject` function and the `Set` keyword. The second argument in the `GetObject` function is the starting object.

Example A.4. *Starting HYSYS*

```
Dim hyCase As Object  
Set hyCase = GetObject("c:\samples\c-2.hsc",  
"HYSYS.SimulationCase")
```

Example A.5 is identical to previous one except that the object variable `hyCase` is declared using the actual object name as it appears in the type library. This assumes that a reference to the type library has already been set.

Example A.5. *Starting HYSYS*

```
Dim hyCase As SimulationCase  
Set hyCase = GetObject("c:\samples\c-2.hsc",  
"HYSYS.SimulationCase")
```

Example A.6 uses early binding in the object declaration. The `CreateObject` command is used to bring up an instance of HYSYS. The starting object here is the `HYSYS.Application` object since a case is not initially being opened. The `SimulationCases` object is accessed through the `Application` object and the `Open` method of `SimulationCases` is used to bring up a specific HYSYS case. The `hyCase` object is set to the active case through the

ActiveDocument property of hyApp. This syntax has been used in the hybrid application.

Example A.6. Starting HYSYS

```
Dim hyCase As SimulationCase  
Dim hyApp As HYSYS.Application  
Set hyApp = CreateObject("HYSYS.Application")  
hyApp.SimulationCases.Open("c:\hysys\samples\c-2.hsc")  
Set hyCase = hyApp.ActiveDocument
```

The sequence of objects is set through a special *dot function*. Properties and methods for an object are also accessed through the *dot function*. It is preferable to keep the sequence of objects to a minimum since each dot function is a call to link between the client and the server application.

The object hierarchy is an important and fundamental concept for utilizing Automation. A particular property can only be accessed by following a specific chain of objects. The chain *always* begins with either the Application or SimulationCase object and ends with the object containing the desired property.

The methods of an object are accessed in the same fashion as properties by utilizing the *dot function*. A method for a particular object is nothing more than a function or subroutine whose behavior is related to the object in some fashion. Typically, the methods of an object will require arguments to be passed when the method is called. The type library will provide information about which arguments are necessary to call a particular method. A function will return a value.

Example A.7 starts up HYSYS and opens a specific case. The temperature value of a specific stream is then obtained. The temperature value is obtained through a connection of three objects; SimulationCase, Flowsheet, and MaterialStreams.

Example A.7. *Accessing HYSYS object properties*

```
Dim hyCase As SimulationCase
Dim TempVal As Double
Set hyCase = GetObject("c:\c-2.hsc", "HYSYS.SimulationCase")
TempVal =
hyCase.Flowsheet.MaterialStreams.Item(0).TemperatureValue
MsgBox TempVal
```

Example A.8 also accesses the temperature value of a specific stream but creates some intermediate objects so that when the temperature value is actually requested the chain of objects only contains one object.

Example A.8. *Accessing HYSYS object properties*

```
Dim hyCase As SimulationCase
Dim hyFlowsheet As Flowsheet
Dim hyStream As ProcessStream
Dim TempVal As Double
Set hyCase = GetObject("c:\samples\c-2.hsc",
"HYSYS.SimulationCase")
Set hyFlowsheet = hyCase.Flowsheet
Set hyStream = hyCase.Flowsheet.MaterialStreams.Item(0)
TempVal = hyStream.TemperatureValue
MsgBox TempVal
```

4 HYSYS Basis objects

The Basis objects refer predominantly to objects handled by the HYSYS BasisManager. The BasisManager object in HYSYS is responsible for handling the global aspects of a HYSYS simulation case. These objects include reactions, components, and property packages. The BasisManager object is accessed through the SimulationCase object. From the BasisManager object, the FluidPackages and HypoGroups collection objects are accessed. Changing the objects accessed directly or indirectly through the BasisManager such as FluidPackages, PropertyPackage, Components, and Hypotheticals requires notification to the HYSYS simulation environment. The BasisManager object contains methods that allow changes to the basis to take place smoothly. The following methods must be used on the outer limits of any code that makes changes to Basis objects. **Examples A.9 and A.10** show the syntax for changing Basis values.

Example A.9. Syntax for changing Basis values

```
SimulationCase.BasisManager.StartBasisChange
... changes to components, interaction parameters, etc.
SimulationCase.BasisManager.EndBasisChange
```

Example A.10. Syntax for changing Basis values

From the BasisManager

```
SimulationCase.BasisManager.FluidPackages.Item(0)
```

From the Flowsheet

```
SimulationCase.Flowsheet.FluidPackage
```

Although both examples of syntax shown above lead to a FluidPackage object, there are differences that exist which are only apparent when attempting to use the Fluid Package. The FluidPackages object returned by the BasisManager object is a collection object containing all FluidPackage objects in a case. Each FluidPackage object can have its own PropertyPackage object and Components object. When the Fluid Package is accessed in this way, changes can be made to the Property Package and the list of Components.

When obtaining a reference to the FluidPackage object from the Flowsheet object, the one Fluid Package associated with the Flowsheet is being accessed. The Property Package or Component list of the FluidPackage object may be viewed, however no changes may be made.

Example A.11 displays the number of FluidPackage objects in a case and sets a FluidPackage object to the first member of the FluidPackages collection.

Example A.11. FluidPackage

```
Dim hyFluidPackages As FluidPackages
Dim hyFluidPackage As FluidPackage
Set hyFluidPackages = hyCase.BasisManager.FluidPackages
MsgBox "Number of Fluid Packages = " & hyFluidPackages.Count
Set hyFluidPackage = hyFluidPackages.Item(0)
```

The PropPackage object is accessed through the FluidPackage object. Each FluidPackage object can have a single PropPackage object. The type of Property Package can be determined through the PropPackage TypeName property or through the PropertyPackageName property of the FluidPackage object. Each Property Package will have a set of unique properties and methods along with the common ones shared among all Property Packages.

Example A.12 connects to the FluidPackages collection object and checks each member FluidPackage to see if it contains the *PengRobinson* Property Package.

Example A.12. *Property Package*

```
Dim hyFluidPackages As FluidPackages
Dim hyFluidPackage As FluidPackage
Dim hyBasis As BasisManager
Dim hyPropPackage As PropPackage
Set hyBasis = hyCase.BasisManager
Set hyFluidPackages = hyBasis.FluidPackages
For Each hyFluidPackage In hyFluidPackages
If hyFluidPackage.PropertyPackageName = "PengRobinson" Then
MsgBox "PengRobinson Property Package is Present"
Set hyPropPackage = hyFluidPackage.PropPackage
End If
Next hyFluidPackage
```

5 Components

The Components object is accessed through the FluidPackage object. Each FluidPackage may have its own unique set of Components.

Example A.13 shows how to access the Components object associated with a particular FluidPackage object. In this example, each component's normal boiling point is checked and a tally of all the components whose boiling point is below 0 °C is counted.

Example A.13. Components

```

Dim numComp as Integer
Dim hyFluidPackage As FluidPackage
Dim hyBasis As BasisManager
Dim hyComponents As Components
Dim hyComponent As Component
Set hyBasis = hyCase.BasisManager
Set hyFluidPackage = hyBasis.FluidPackages.Item(0)
Set hyComponents = hyFluidPackage.Components
numComp = 0
For Each hyComponent In hyComponents
If hyComponent.NormalBoilingPointValue < 0 Then
numComp = numComp + 1
End If
Next hyComponent
MsgBox numComp & " components have NBP below 0"

```

6 Fluid

A Fluid object is derived from a single ProcessStream through the DuplicateFluid method. A Fluid object is essentially an internal copy of the ProcessStream that can be manipulated for property calculation purposes. The ProcessStream and Fluid share many of the same properties. A Fluid however can be flashed without interfering with the simulation case.

Example A.14 shows how to create a Fluid out of a stream and use the Fluid to perform a flash calculation. The DuplicateFluid method returns a Fluid object. A variety of flashes could have been performed, but in this case, a pressure vapor-fraction flash is run with the desired pressure and vapor-fraction used as arguments to the method.

Example A.14. Fluid

```

Public Sub FluidExample(pressureval As Double)
Dim hyFluid As Fluid
Dim hyStream As ProcessStream
Set hyStream = hyCase.Flowsheet.MaterialStreams.Item(0)
Set hyFluid = hyStream.DuplicateFluid
hyFluid.PVFlash pressureval, 0
MsgBox "Bubble Point Temperature = " &
hyFluid.TemperatureValue
End Sub

```


Appendix B

Working with LinkControl

This appendix explains working procedure of LinkControl executive program. Main user interfaces that are applied to interact with the program are shown and explained. Important points that should be observed while preparing the data for use in the hybrid application have also been discussed.

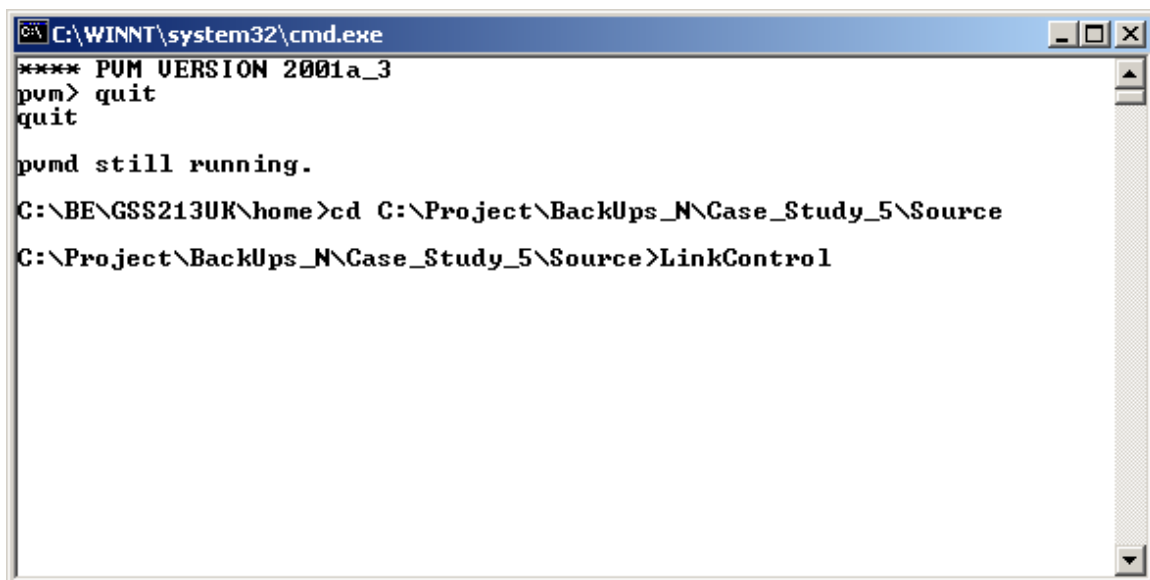
1 Required simulators

Performing simulation using the hybrid application precludes installation of GeoQuest Simulation Suite and Hyprotech's HYSYS on the system. GeoQuest software is now available through GeoQuest Simulation Software Launcher that includes all GeoQuest products. Eclipse 100 and 300 are available through this application as well.

2 Initializing PVM

Parallel Virtual Machine (PVM) is the protocol by which Eclipse communicates with the rest of the system. Consequently, PVM should be started before any communication can be done. PVM can be loaded both through GeoQuest Launcher and by running \$PVM batch file on a DOS command prompt.

After PVM is started, using `quit` command, DOS prompt becomes available on the DOS window. In the next step, the current directory is changed to the directory where LinkControl executable has been stored and LinkControl is run. It is important to run LinkControl on the same DOS window created by PVM or the one on which \$PVM was executed. Otherwise, PVM fails to locate the Eclipse executable. All the above steps have been depicted on **Figure B.1**.



```
C:\WINNT\system32\cmd.exe
**** PVM VERSION 2001a_3
pvm> quit
quit

pvm still running.

C:\BE\GSS213UK\home>cd C:\Project\BackUps_N\Case_Study_5\Source
C:\Project\BackUps_N\Case_Study_5\Source>LinkControl
```

FIGURE B.1 Required steps to initialize simulation using hybrid application.

3 Using LinkControl

After running LinkControl, the dialog box shown on **Figure B.2** appears. Using the Launcher dialog box, user can locate the Eclipse simulation data file, and two HYSYS simulation cases belonging to network and topside facility, respectively.

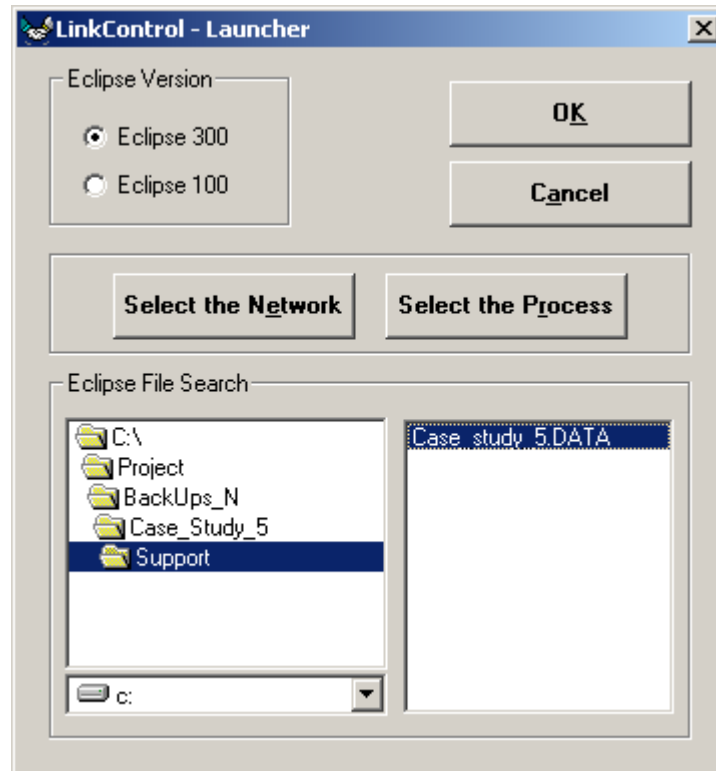


FIGURE B.2 LinkControl launcher dialog box.

After the required files have been loaded, the main LinkControl dialog box appears. On this dialog depicted in **Figure B.3**, user should perform the following:

1. From the material streams list box, select the material stream that should be connected to wells simulated in Eclipse. The selected streams with correct order will appear on the list box to the right of the dialog. It is advisable to check this list to make sure that correct streams have been connected to the wells.

2. Select the network outlet stream that will be connected to process feed stream using dropdown lists in the Linking HYSYS cases frame.
3. Select the required time-step for reservoir calculations.
4. Call the reservoir simulator.

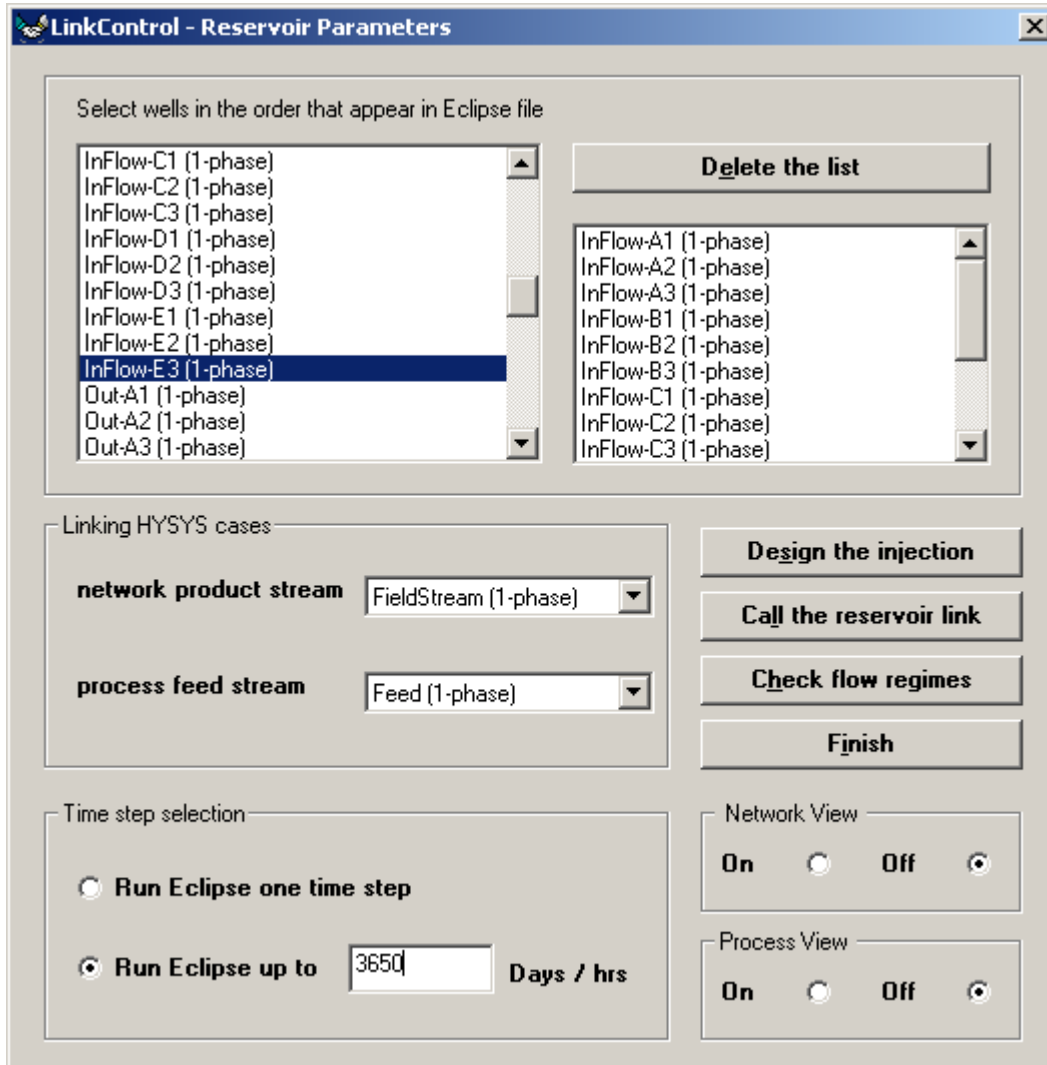
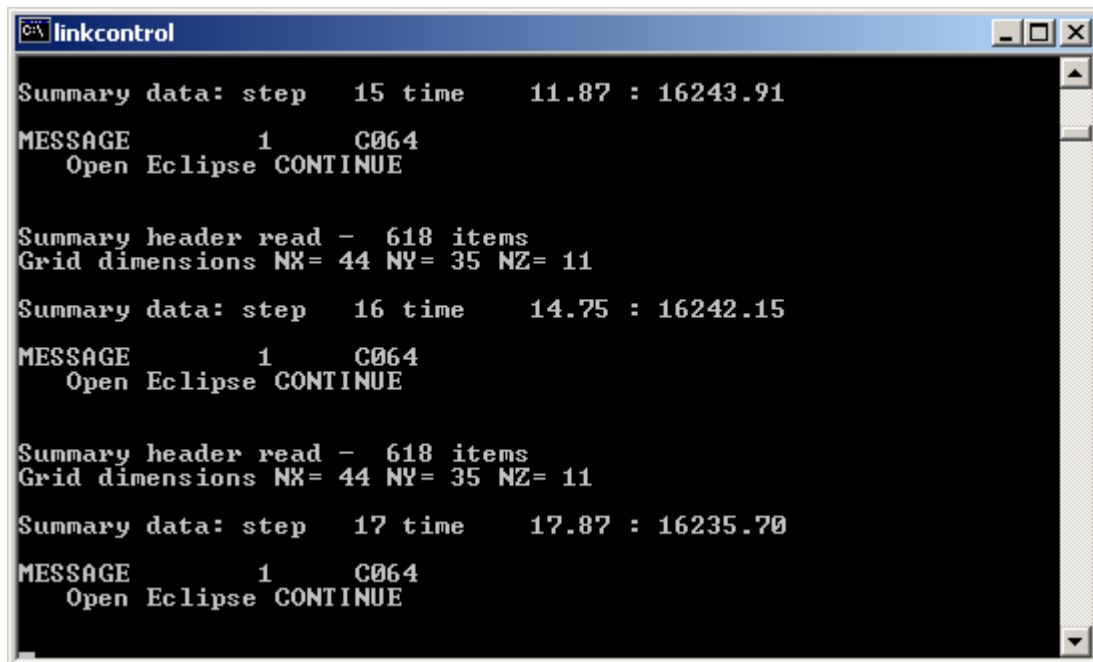


FIGURE B.3 Main LinkControl dialog.

The reservoir simulation begins at this step and continues up to the time selected by the user. Simulation output results and messages from Open-Eclipse appear on the screen as shown on **Figure B.4**. After reservoir

simulation is finished, the results are automatically sent to HYSYS network simulation case. HYSYS calculations to solve the network begins automatically. After the network is solved, the network outlet stream is copied to the process simulation case as the feed stream and calculation of the process side begins. The overall simulation terminates after the process simulation case has finished its calculations.

At this stage, the simulation results are available for processing. It is possible to save the solved network and process simulation cases through LinkControl.



```
linkcontrol
Summary data: step 15 time 11.87 : 16243.91
MESSAGE 1 C064
Open Eclipse CONTINUE

Summary header read - 618 items
Grid dimensions NX= 44 NY= 35 NZ= 11

Summary data: step 16 time 14.75 : 16242.15
MESSAGE 1 C064
Open Eclipse CONTINUE

Summary header read - 618 items
Grid dimensions NX= 44 NY= 35 NZ= 11

Summary data: step 17 time 17.87 : 16235.70
MESSAGE 1 C064
Open Eclipse CONTINUE
```

FIGURE B.4 Open-Eclipse output messages printed to screen.

4 Data preparation for the hybrid application

There are a number of points that should be observed when input data is prepared for use in the hybrid application. The points to observe are:

1. The number of wells simulated in Eclipse should be exactly the same as the inflow streams considered in the network simulation case.
2. The same Property Package should be used in network and process simulation cases, otherwise the network results can not be used by the process simulation case.
3. The sequence of the components defined in Eclipse and HYSYS data files should be the same with water as the last component. Different names however can be used for components in Eclipse and HYSYS simulations.
4. Reservoir temperature is not inquired by LinkControl from Eclipse, therefore, the same temperature used in the Eclipse data file should be used to define the temperature for the inflow streams. Rest of the temperatures are calculated by HYSYS.
5. The reservoir depth defined in the reservoir data file, should correspond to the well length used in the network simulation case.
6. After simulation is finished, the PVM command prompt should be loaded by executing \$PVM batch file. The PVM should subsequently be terminated by entering `halt` command at the PVM command prompt.

Appendix C

Flow regime determination

This appendix discusses two-phase flow regime determination in compositional systems using a unified flow regime determination routine. This routine is the basis of DoFazi program used in this study.

The use of DoFazi program when loaded through LinkControl has been demonstrated as well.

1 Flow regime determination routine

In this section, after a brief introduction to two-phase flow and defining the concept of flow pattern, a unified flow pattern determination model that has been used in DoFazi program will be introduced.

2 Brief introduction to two-phase flow

A two-phase flow system can be defined as a system in which a gas and a liquid phase flow together under various flow situations known as flow regime. When two phases flow simultaneously in pipes, the flow behavior is much more complex than the cases where a single phase, either gas or liquid, flows in the conduit. In two-phase flow, the phases tend to separate due to differences in density. Shear stresses at the pipe wall are different for each phase as a result of their different densities and viscosities. Expansion of highly compressible gas phase with decreasing pressure increases the local volumetric flow rate of the gas. As a result, the gas and liquid phases normally do not travel at the same velocity in the pipe. For upward flow, the less dense, more compressible, less viscous gas phase tends to flow at higher velocity than the liquid phase, causing a phenomenon known as slippage. However, for downward flow the liquid often flows faster than the gas.

Perhaps the most distinguishing characteristic of two-phase flow is the variation in the physical distribution of the phases, a characteristic known as flow regime or flow pattern. During two-phase flow through pipes, the flow pattern that exists depends on the relative magnitudes of the forces that act on the fluids. Buoyancy, turbulence, inertia and surface tension forces vary significantly with flow rates, pipe diameter, inclination angle, and fluid properties of the phases. Several different flow patterns can exist in a given pipeline as a result of the large pressure and temperature changes that the fluids encounter. Especially important is that the pressure gradient varies significantly with flow pattern. Thus, the ability to predict flow pattern as a function of the flow parameters is of primary concern.

The increased complexity of multiphase flow has logically resulted in a higher degree of empiricism for predicting flow behavior. Many empirical correlations have been developed for predicting flow pattern, slippage between phases, friction factors and pressure drop for two-phase flow in pipes. However, since the mid 1970's dramatic improvements in

understanding the fundamental mechanisms that govern two-phase flow have taken place. These have resulted in new predictive methods that rely much less on empirical correlations.

3 Flow patterns in two-phase systems

For vertical two-phase flow of gas and liquid, most investigators now recognize the existence of four flow patterns (Taitel, 1980): bubble flow, slug flow, churn flow, and annular flow. These flow patterns shown schematically in **Figure C.1**, are described below. Slug and churn flow are sometimes combined into a flow pattern called intermittent flow. It is also common to introduce a transition between slug flow and annular flow that incorporates churn flow (Aziz, 1972). Finally, some investigators have named annular flow as mist or annular-mist flow.

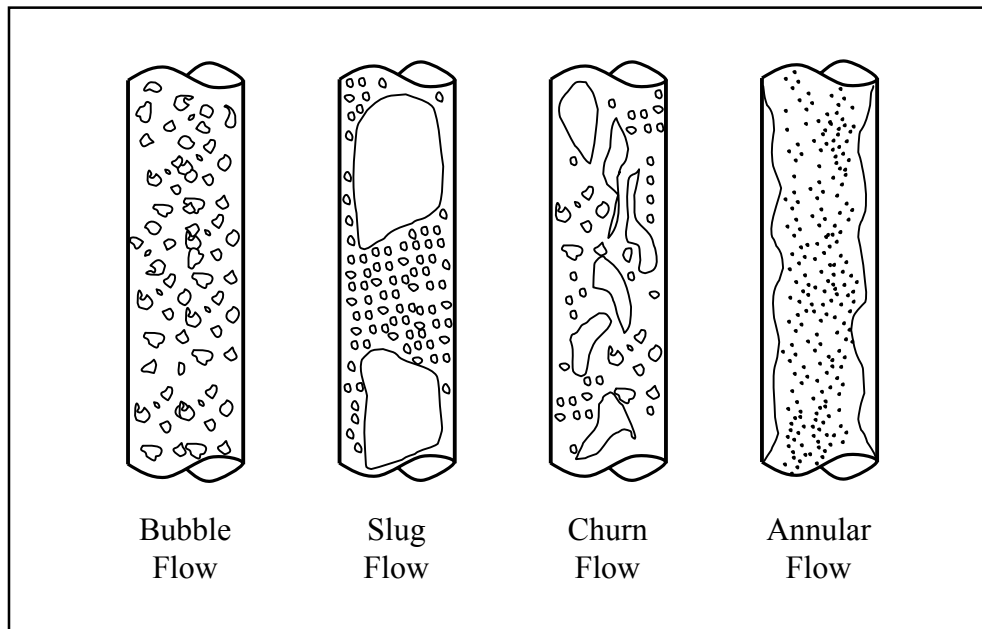


FIGURE C.1 Vertical flow patterns.

3.1 Bubble flow

Bubble flow is characterized by a uniformly distributed gas phase as discrete bubbles in a continuous liquid phase. Based on the presence or absence of relative velocity between the two phases, bubble flow is further classified into bubbly and dispersed bubble flows. In bubbly flow, relatively fewer and

larger bubbles move faster than the liquid phase due to the relative velocity. On the other hand, in dispersed bubble flow, numerous tiny bubbles are transported by the liquid phase, causing no relative motion between the two phases.

3.2 Slug flow

Slug flow is characterized by a series of slug units, each of which is composed of a gas pocket called a Taylor bubble, a plug of liquid called a slug and a film of liquid around the Taylor bubble flowing downwards.

3.3 Churn flow

Churn flow is a chaotic flow of gas and liquid in which both the Taylor bubbles and the liquid slugs are distorted in shape. Neither of the phases appears to be continuous.

3.4 Annular flow

Annular flow is characterized by axial continuity of the gas phase in a central core with the liquid flowing, both as a thin film along the pipe wall and as dispersed droplets in the core. At high gas flow rates, more liquid becomes dispersed in the core, leaving a very thin liquid film flowing along the wall.

As is shown in **Figure C.2**, for horizontal flow, four basic flow regimes have been considered (Beggs, 1973): stratified flow, intermittent flow, annular flow, and dispersed bubble flow. Except for stratified flow that is unique to horizontal and near horizontal flow situation, the rest of the flow patterns are quite similar in nature to their counterparts in vertical flow and are not explained again.

3.5 Stratified flow

Stratified flow is characterized by separate gas and liquid phases flowing together in two, well-defined flow zones usually at different velocities. Stratified flow is subdivided into stratified smooth and stratified wavy flow patterns. The former pertains to lower gas superficial velocities and the later prevails when gas superficial velocity increases causing increased shear between the gas and liquid phases at the phase interface.

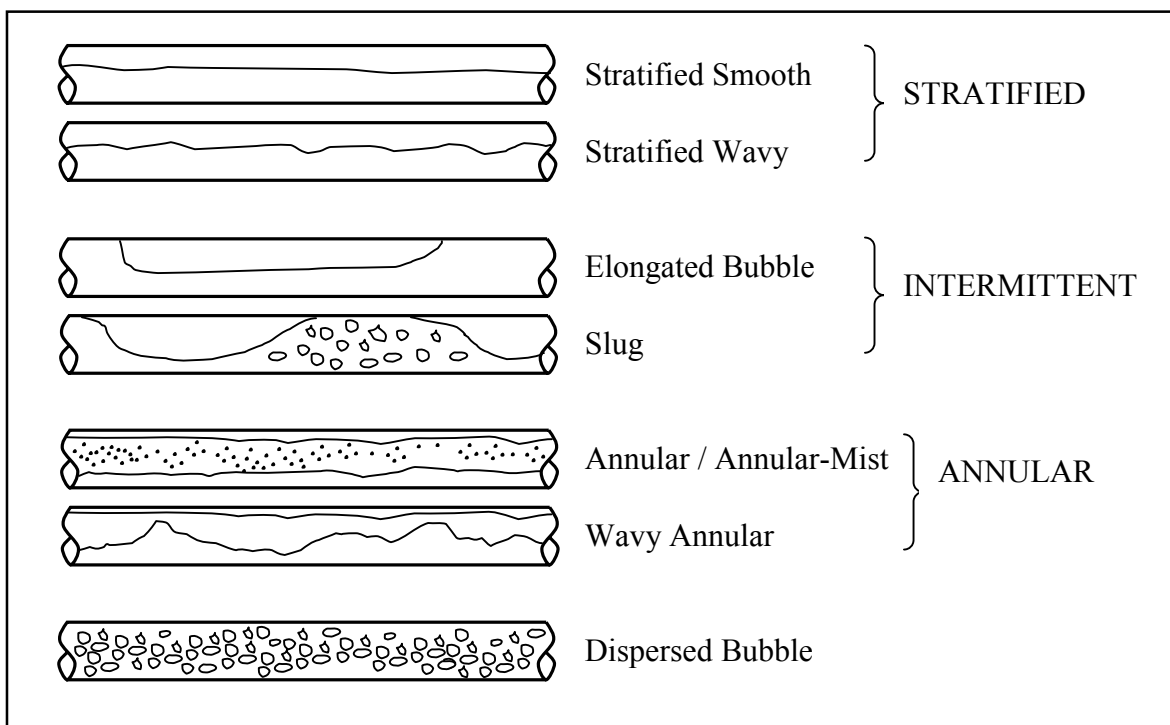


FIGURE C.2 Horizontal flow patterns.

In this study, a unified model for predicting flow-pattern transitions over the whole range of pipe inclinations compiled by Barnea (1987) has been used to determine the flow regime. The Barnea's unified model is considered to be one of the most general methods that allow the prediction of flow pattern, once the flow rates, the conduit geometry, the inclination angle and the fluid properties are specified. Barnea's model has been explained in details in this appendix.

4 Black oil versus compositional model

With the exception of few, nearly all two-phase pipeline simulations have been traditionally performed using black oil simulators. The black oil model is described by three basic parameters; solution gas oil ratio, R_s , oil formation volume factor, B_o , and gas formation volume factor, B_g . A black oil model's validity rests on the assumption that the hydrocarbon mixture is composed of only two components, denoted oil and gas, each with fixed composition (Gould, 1979). The gas is said to be dissolved in the oil, with the amount of dissolved gas decreasing with decreasing pressure. The oil component in the pipeline generally is defined as stock-tank oil. A black oil

model usually treats main PVT properties, i.e., solution gas oil ratio, densities and viscosities as single-value functions of pressure.

A multi-component or compositional approach is designed for gas condensate and volatile oil systems that are not handled easily using black oil method. These fluids are represented as N -component hydrocarbon mixtures and a single equation of state is used to determine physical properties.

Heavy calculation load of the compositional model is considered to be the most important obstacle in using it in applications where the main aim is simply property determination. A relatively huge body of auxiliary information is required to perform flash calculation on a multi-component system. The need for reliable stability test routines that determine whether the composition under consideration forms a two-phase mixture at the specified pressure and temperature adds to the difficulties involved in the use of compositional models. Therefore, the calculation cost of a compositional model discourages its use in situations where the original calculation requires the compositional model for purposes as simple as property calculation.

However, considering the fact that a compositional model is the only reliable model available for systems such as gas condensate or volatile oil, its use in many instances is justified. On the other hand, the compositional model is a more realistic model for most multi-component systems.

If an already available compositional model can be easily integrated into an application that requires the outputs of a compositional model, then all of the mentioned problems are solved. Through Automation techniques, this can be achieved by integrating HYSYS calculation engine to a program written, for instance, in Visual Basic. This was done in DoFazi program. This program uses HYSYS calculation engine to calculate the required phase properties for flow regime determination. DoFazi flashes the material stream for which the flow regime is required at its corresponding pressure and temperature using the method explained in Appendix A. The flow rate, density, and viscosity of each phase and the surface tension in the two-phase mixture are obtained through Automation and used in flow regime determination routine.

5 Flow regime determination model

In this section, a concise description of the Barnea's unified flow regime or flow pattern determination model is presented. Complete explanation of the theories behind the model is considered beyond the scope of this appendix.

The main purpose of the model is to construct a completely general method that allows the prediction of the flow pattern, once the flow rates, the conduit geometry, the inclination angle and the fluid properties are specified. The aim has not been fully achieved by the Barnea's unified model but it is still considered as one of the most general methods with results that agree acceptably with experimental observations.

The method works by discovering various transitions between the patterns that have been accepted by most investigators. In the sections that follow, the criteria for each transition is explained and finally a flow diagram is presented, summarizing the steps needed to be followed in order to determine the flow regime.

5.1 The transition from dispersed bubbles

Dispersed bubble flow usually appears at very high liquid flow rates. There are, however, conditions where small discrete bubbles also appear at low liquid rate. These bubbles are sometimes designated as bubble or bubbly flow. The bubbly flow pattern can exist if two conditions are met:

1. The Taylor bubble velocity exceeds the bubble velocity. This condition is satisfied in large-diameter pipes:

$$D > 19 \left[\frac{(\rho_L - \rho_G)\sigma}{\rho_L^2 g} \right]^{1/2} \quad [1]$$

2. The angle of inclination, β , is large enough to prevent migration of bubbles to the top wall of the pipe:

$$\frac{\cos \beta}{\sin^2 \beta} = \frac{3}{4} \cos 45^\circ \frac{U_0^2}{g} \left(\frac{C_L \gamma^2}{d} \right) \quad [2]$$

where U_0 is:

$$U_0 = 1.53 \left[\frac{g(\rho_L - \rho_G)\sigma}{\rho_L^2} \right]^{1/4} \quad [3]$$

The average value suggested for C_L is 0.8 and γ ranges from 1.1 to 1.5 based on observation.

When these two conditions are satisfied, bubbly flow is observed even at low rates were turbulent forces do not cause bubble breakup.

The transition from bubbly to slug flow takes place when the gas void fraction exceeds a critical value of $\alpha_c = 0.25$. This transition is given by:

$$U_{LS} = \frac{1-\alpha}{\alpha} U_{GS} - 1.53(1-\alpha) \left[g \frac{(\rho_L - \rho_G)\sigma}{\rho_L^2} \right]^{1/4} \sin \beta \quad [4]$$

where $\alpha = \alpha_c = 0.25$. The angle β is positive for upward flow and negative for downward flow.

The modified criterion that accounts for pipe inclination for transition mechanism from the dispersed bubbles for upward vertical flow is:

$$d_c \geq \left[0.75 + 4.15 \left(\frac{U_{GS}}{U_M} \right)^{1/2} \right] \left(\frac{\sigma}{\rho_L} \right)^{3/5} \left(\frac{2f_M U_M^3}{D} \right)^{-2/5} \quad [5]$$

The bubble diameter on the transition boundary, d_c , is a function of the liquid velocity and the angle of inclination. The value of d_c is taken as the smallest between d_{CD} and d_{CB} , where d_{CD} is the critical bubble size above which the bubble is deformed,

$$d_{CD} = 2 \left[\frac{0.4\sigma}{(\rho_L - \rho_G)g} \right]^{1/2} \quad [6]$$

d_{CB} is the critical bubble size below which migration of bubbles to the upper part of the pipe is prevented,

$$d_{CB} = \frac{3}{8} \left[\frac{\rho_L}{\rho_L - \rho_G} \right] \frac{f_M U_M^2}{g \cos \beta} \quad [7]$$

The transition boundary [5] is valid for $0 \leq \alpha \leq 0.52$. At the upper limit, the maximum volumetric packing density of the bubbles is reached and coalescence occurs even at high turbulent levels. The transition curve that characterizes this condition is:

$$U_{LS} = U_{GS} \frac{1 - \alpha}{\alpha} \quad [8]$$

where $\alpha = 0.52$.

5.2 The stratified non-stratified transition

For horizontal and slightly inclined tubes, the transition from equilibrium stratified flow is believed to be due to Kelvin-Helmholtz instability, which considers stratified flow with a finite wave on the surface over which the gas flows. As the gas accelerates over the wave crest, the pressure in the gas phase decreases due to the Bernoulli effect and the wave tends to grow. On the other hand, the force of gravity acting on the wave tends to cause it to decay. The situation at which the wave will grow and the transition from stratified flow occurs is:

$$F^2 \left[\frac{1}{(1 - \tilde{h}_L)^2} \frac{\tilde{U}_G^2 \frac{d\tilde{A}_L}{d\tilde{h}_L}}{\tilde{A}_G} \right] \geq 1 \quad [9]$$

where F is the Froude number modified by the density ratio:

$$F = \sqrt{\frac{\rho_G}{\rho_L - \rho_G}} \frac{U_{GS}}{\sqrt{Dg \cos \beta}} \quad [10]$$

The dimensionless variables are defined by:

$$\begin{aligned}\tilde{h}_L &= \frac{h_L}{D} \\ \tilde{A}_L &= \frac{A_L}{D^2} \\ \tilde{A}_G &= \frac{A_G}{D^2} \\ \tilde{U}_L &= \frac{U_L}{U_{LS}} = \frac{A}{A_L} \\ \tilde{U}_G &= \frac{U_G}{U_{GS}} = \frac{A}{A_G}\end{aligned}\quad [11]$$

h_L is the liquid level in equilibrium stratified flow and can be determined from the Martinelli parameters X and Y defined in **Equations [12a]** and **[12b]** below. Once X and Y have been calculated, h_L / D can be determined from **Figure C.3**.

$$X^2 = \frac{\frac{4}{D} f_{LS} \frac{\rho_L U_{LS}^2}{2}}{\frac{4}{D} f_{GS} \frac{\rho_G U_{GS}^2}{2}} = \frac{\left(\frac{dp}{dx}\right)_{LS}}{\left(\frac{dp}{dx}\right)_{GS}} \quad [12a]$$

$$Y = \frac{(\rho_L - \rho_G)g \sin \beta}{\left(\frac{dp}{dx}\right)_{GS}} \quad [12b]$$

(dp/dx) designates the pressure drop of one phase superficially flowing alone in the pipe.

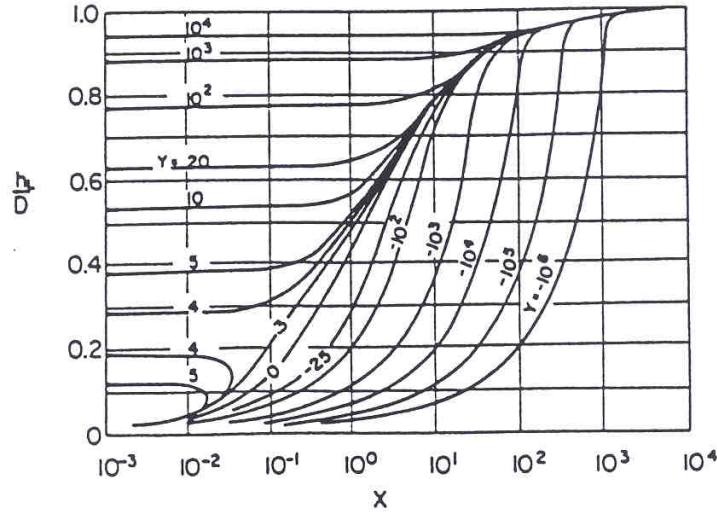


FIGURE C.3 Equilibrium liquid level in stratified flow.

5.3 The stratified annular transition

Transition mechanism [9] presents a criterion under which finite waves on stratified liquid flow are expected to grow resulting in either slug or annular flow. However, at steep downward inclinations, another mechanism comes into play, by which stable stratified flow is seen to change into annular flow at relatively low gas flow rates.

At high downward inclination angles, the stratified liquid level is small and the liquid velocity U_L is very high. Under these conditions, droplets are torn from the wavy turbulent interface and may be deposited on the upper wall, resulting in an annular film. The condition for this type of annular flow to take place is:

$$U_L^2 \geq \frac{gD(1 - \tilde{h}_L)\cos \beta}{f_L} \quad [13]$$

or in dimensionless form:

$$Z = \frac{\left(\frac{dp}{dx}\right)_{LS}}{\rho_L g \cos \beta} \geq 2 \left(\frac{\tilde{A}_L}{\tilde{A}}\right)^2 (1 - \tilde{h}_L) \frac{f_{LS}}{f_L} \quad [14]$$

5.4 The transition from annular to intermittent flow

This transition can be assumed to occur when the gas core is blocked at any location by the liquid. Blockage of the gas core may result from two possible mechanisms:

1. Instability of the liquid film, due to partial downward flow of liquid near the wall, causing blockage at the entrance.
2. Blockage of the gas core as a result of large supply of liquid in the film.

The condition for the stability of the liquid film in annular flow [mechanism (a)] is obtained from the simultaneous solution of the following dimensionless equations:

$$Y = \frac{1 + 75\alpha_L}{(1 - \alpha_L)^{5/2} \alpha_L} - \frac{1}{\alpha_L^3} X^2 \quad [15]$$

and

$$Y \geq \frac{2 - \frac{3}{2}\alpha_L}{\alpha_L^3 \left(1 - \frac{3}{2}\alpha_L\right)} X^2 \quad [16]$$

X and Y were defined in **Equations [12 a,b]**. **Equation [15]** gives the steady state solution for the liquid holdup, α_L , in the annular flow. The value of α_L that satisfies the condition for the film instability is obtained from **[16]**.

Blockage of the gas core by liquid lumps [mechanism (b)] will occur when the liquid supply in the film is large enough to provide the liquid needed to bridge the pipe. The condition for slugging is:

$$\frac{A_L}{A \cdot R_{sm}} = \frac{\alpha_L}{R_{sm}} \geq 0.5 \quad [17]$$

where R_{sm} is the minimum liquid holdup within the formed liquid slug that will allow complete bridging of the gas passage.

5.5 Sub-regions within intermittent flow

The intermittent pattern is usually subdivided into elongated bubble, slug and churn flows. Basically, these three flow patterns have the same configuration with respect to the distribution of the gas and the liquid interfaces. In these flow patterns, slugs of liquid are separated by large bullet-shaped bubbles. In slug flow, the liquid bridges are aerated by small gas bubbles. The elongated bubble pattern is considered the limiting case of slug flow when the liquid slug is free of entrained bubbles, while the churn flow takes place when the gas void fraction within the liquid slug reaches a maximum value above which occasional collapse of the liquid slugs occurs.

The gas holdup on the transition line from the dispersed bubbles is the maximum holdup that the liquid slug can accommodate as fully dispersed bubbles at a given velocity $U_M = U_{GS} + U_{LS}$. Thus, curves of constant U_M within the intermittent region represent the locus where α_s is constant and is equal to the holdup of the dispersed bubble pattern at the transition boundary. Once the fluid properties and pipe size are set, α_s can be obtained from [5] to yield:

$$\alpha_s = 1 - R_s = 0.058 \left[d_c \left(\frac{2f_M}{D} U_M^3 \right)^{2/5} \left(\frac{\rho_L}{\sigma} \right)^{3/5} - 0.725 \right]^2 \quad [18]$$

when $\alpha_s = 0$ then the elongated bubble-slug transition is obtained. When the gas holdup within the liquid slug reaches the maximum bubble volumetric packing ($\alpha_s = 0.52$), the continuity of the very aerated liquid slug is destroyed by bubble agglomeration and the formation of regions of high gas concentration within the liquid slug, resulting in transition to churn flow.

5.6 Sub-regions in the stratified flow

Two sub-regions are usually defined in the stratified flow: stratified smooth and stratified wavy. Waves may form on a smooth liquid interface due to the action of the gas flowing over the liquid (typical of horizontal tubes) or as a result of the action of gravity, even in the absence of gas flow (typical in downward inclined pipes).

The following condition for wave generation as a result of the “wind effect” has been suggested:

$$U_G \geq \left[\frac{4\nu_L(\rho_L - \rho_G)\rho \cos \beta}{s\rho_G U_L} \right]^{1/2} \quad [19]$$

or in dimensionless form:

$$K \geq \left[\frac{2}{\tilde{U}_G \sqrt{\tilde{U}_L} \sqrt{s}} \right] \quad [20]$$

where ν_L is the liquid kinematic viscosity, s is a sheltering coefficient and K is the product of the modified Froude number and the square root of the superficial Reynolds number of the liquid:

$$K^2 = F^2 \text{Re}_{LS} = \left[\frac{\rho_G U_{GS}^2}{(\rho_L - \rho_G) D g \cos \beta} \right] \left[\frac{D U_{LS}}{\nu_L} \right] \quad [21]$$

As was mentioned, waves can develop on liquid flowing downward slopes, even in the absence of interfacial shear with gas flow. For turbulent flow in smooth pipes the criteria for wave inception is:

$$\text{Fr} = \frac{U_L}{\sqrt{gh_L}} \geq 1.5 \quad [22]$$

$$W = \frac{U_{LS}}{\sqrt{gD}} \geq 1.5 \sqrt{\tilde{h}_L} \frac{\tilde{A}_L}{\tilde{A}} \quad [23]$$

A logical path for the systematic determination of the various flow patterns is presented by the flow chart shown in **Figure C.4**.

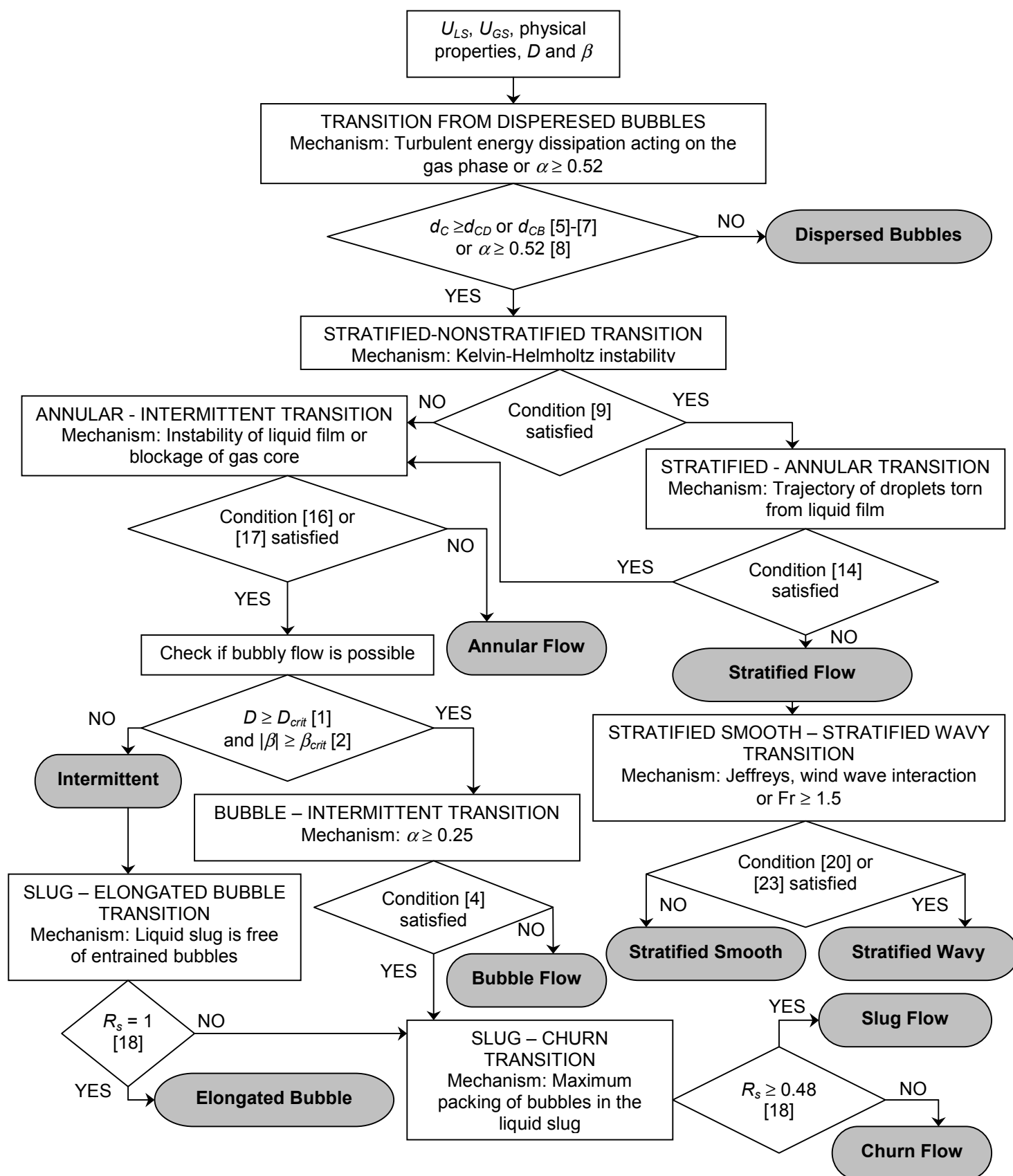


FIGURE C.4 Logical pass for flow regime determination.

6 Working with flow regime determination model

Based on the unified model explained in previous section, the DoFazi program was developed and integrated with the hybrid application to study the existing flow regimes in cases under study. DoFazi can be run through LinkControl to determine the existing flow regimes in simulation cases that have been loaded by LinkControl.

The dialog box shown in **Figure C.5** appears after DoFazi is executed through LinkControl using the dialog shown in **Figure B.3**. The list box on the flow regime dialog shows a list of all two-phase material streams that exist in the simulation cases loaded by LinkControl. The user selects the material stream for which the flow regime should be determined and supplies the inside diameter, roughness, and deviation angle of the pipe in which the material stream is flowing. After running DoFazi, the flow regime and phase superficial velocities appear on the dialog box as shown on **Figure C.5**.

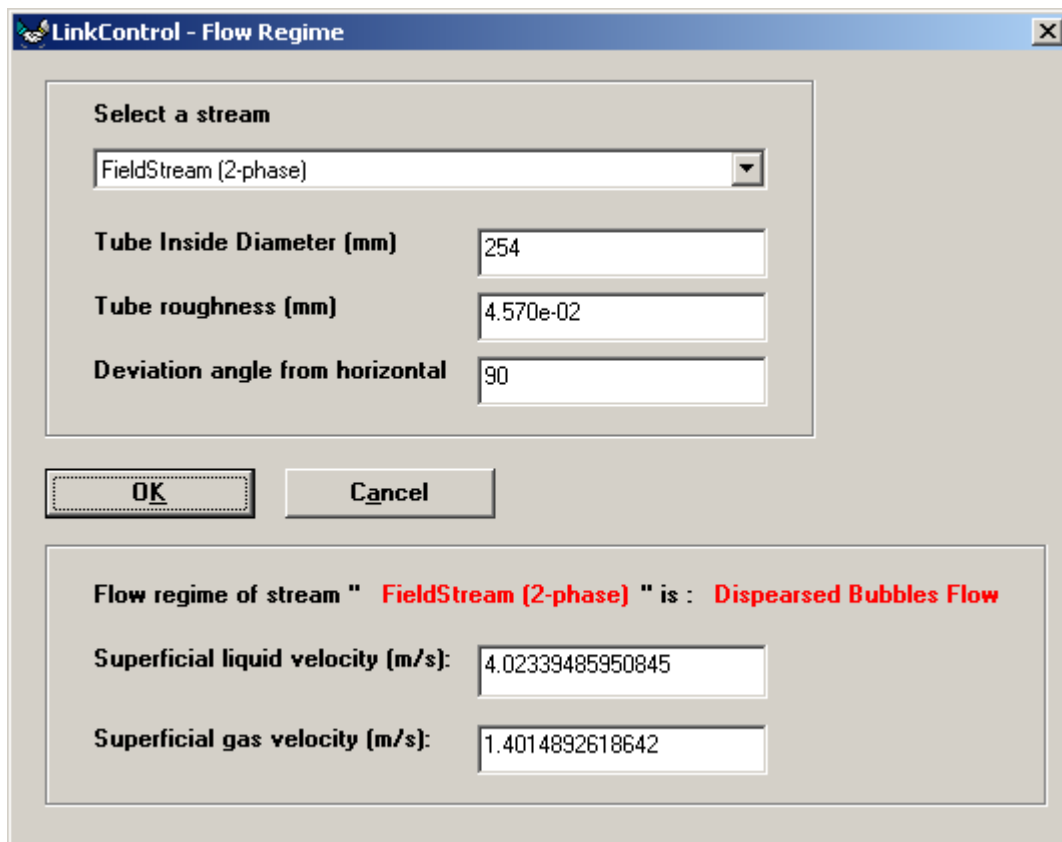


FIGURE C.5 DoFazi dialog box while running through LinkControl.

Appendix D

HYSYS simulation results

This appendix reports selected HYSYS simulation results for **Example 2.1**. The calculation results for two material streams, one well, one pipeline, and one Adjust operation were selected to demonstrate the HYSYS output.

Material Stream: Inflow-2 (1-phase)			Fluid Package:	Basis-2		
			Property Package:	Peng Robinson		
CONDITIONS						
	Overall	Vapour Phase				
Vapour / Phase Fraction	1.0000	1.0000				
Temperature: (C)	200.0	200.0				
Pressure: (kPa)	1.621e+004	1.621e+004				
Molar Flow (kgmole/h)	833.6	833.6				
Mass Flow (kg/h)	1.893e+004	1.893e+004				
Liquid Volume Flow (m3/h)	50.40	50.40				
Molar Enthalpy (kJ/kgmole)	-8.176e+004	-8.176e+004				
Molar Entropy (kJ/kgmole-C)	164.9	164.9				
Heat Flow (kJ/h)	-6.816e+007	-6.816e+007				
Std Liq Volume Flow (m3/h)	---	---				
PROPERTIES						
	Overall	Vapour Phase				
Vapour / Phase Fraction	1.0000	1.0000				
Temperature: (C)	200.0	200.0				
Pressure: (kPa)	1.621e+004	1.621e+004				
Actual Volume Flow (m3/h)	190.7	190.7				
Mass Enthalpy (kJ/kg)	-3601	-3601				
Mass Entropy (kJ/kg-C)	7.261	7.261				
Molecular Weight	22.71	22.71				
Molar Density (kgmole/m3)	4.372	4.372				
Mass Density (kg/m3)	99.27	99.27				
Std Liquid Mass Density (kg/m3)	---	---				
Molar Heat Capacity (kJ/kgmole-C)	66.98	66.98				
Mass Heat Capacity (kJ/kg-C)	2.950	2.950				
Thermal Conductivity (W/m-K)	6.478e-002	6.478e-002				
Viscosity (cP)	2.059e-002	2.059e-002				
Surface Tension (dyne/cm)	---	---				
Z Factor	0.9424	0.9424				
Molar Vapour Fraction	1.0000	1.0000				
Mass Vapour Fraction	1.0000	1.0000				
Volume Vapour Fraction	1.0000	1.0000				
Molar Volume (m3/kgmole)	0.2287	0.2287				
Actual Gas Flow (ACT_m3/h)	190.7	190.7				
Actual Liquid Flow (m3/s)	---	---				
Std. Gas Flow (STD_m3/h)	1.971e+004	1.971e+004				
Std. Liquid Volume Flow (m3/h)	---	---				
Watson K	17.14	17.14				
Kinematic Viscosity (cSt)	0.2074	0.2074				
Cp/Cv	1.306	1.306				
Lower Heating Value (kJ/kgmole)	---	---				
Mass Lower Heating Value (kJ/kg)	---	---				
Liquid Fraction	0.0000	0.0000				
Partial Pressure (kPa)	152.8	152.8				
COMPOSITION						
Overall Phase			Vapour Fraction	1.0000		
COMPONENTS	MOLAR FLOW (kgmole/h)	MOLE FRACTION	MASS FLOW (kg/h)	MASS FRACTION	LIQUID VOLUME FLOW (m3/h)	LIQUID VOLUME FRACTION
Nitrogen	5.7915	0.0069	162.2367	0.0086	0.2012	0.0040
CO2	7.8600	0.0094	345.9178	0.0183	0.4191	0.0083
Methane	602.5367	0.7228	9666.4354	0.5107	32.2867	0.6406
Ethane	38.9802	0.0468	1172.1316	0.0619	3.2954	0.0654
Propane	138.8576	0.1666	6123.2049	0.3235	12.0850	0.2398
i-Butane	2.4341	0.0029	141.4808	0.0075	0.2518	0.0050
n-Butane	6.7541	0.0081	392.5736	0.0207	0.6731	0.0134
i-Pentane	1.5297	0.0018	110.3685	0.0058	0.1770	0.0035
n-Pentane	1.9661	0.0024	141.8579	0.0075	0.2253	0.0045
n-Hexane	2.5255	0.0030	217.6391	0.0115	0.3284	0.0065
Hyprotech Ltd.			HYSYS.Plant v2.2.2 (Build 3806)		Page 1 of 8	

Material Stream: Inflow-2 (1-phase) (continued)					Fluid Package:	Basis-2
					Property Package:	Peng Robinson
COMPOSITION						
Overall Phase (continued)					Vapour Fraction	1.0000
COMPONENTS	MOLAR FLOW (kgmole/h)	MOLE FRACTION	MASS FLOW (kg/h)	MASS FRACTION	LIQUID VOLUME FLOW (m3/h)	LIQUID VOLUME FRACTION
C7PLUS*	0.1012	0.0001	16.2675	0.0009	0.0196	0.0004
H2O	24.2833	0.0291	437.4668	0.0231	0.4383	0.0087
Total	833.6200	1.0000	18927.5805	1.0000	50.4010	1.0000
Vapour Phase					Phase Fraction	1.0000
COMPONENTS	MOLAR FLOW (kgmole/h)	MOLE FRACTION	MASS FLOW (kg/h)	MASS FRACTION	LIQUID VOLUME FLOW (m3/h)	LIQUID VOLUME FRACTION
Nitrogen	5.7915	0.0069	162.2367	0.0086	0.2012	0.0040
CO2	7.8600	0.0094	345.9178	0.0183	0.4191	0.0083
Methane	602.5367	0.7228	9666.4354	0.5107	32.2867	0.6406
Ethane	38.9802	0.0468	1172.1316	0.0619	3.2954	0.0654
Propane	138.8576	0.1666	6123.2049	0.3235	12.0850	0.2398
i-Butane	2.4341	0.0029	141.4808	0.0075	0.2518	0.0050
n-Butane	6.7541	0.0081	392.5736	0.0207	0.6731	0.0134
i-Pentane	1.5297	0.0018	110.3685	0.0058	0.1770	0.0035
n-Pentane	1.9661	0.0024	141.8579	0.0075	0.2253	0.0045
n-Hexane	2.5255	0.0030	217.6391	0.0115	0.3284	0.0065
C7PLUS*	0.1012	0.0001	16.2675	0.0009	0.0196	0.0004
H2O	24.2833	0.0291	437.4668	0.0231	0.4383	0.0087
Total	833.6200	1.0000	18927.5805	1.0000	50.4010	1.0000
K VALUE						
COMPONENTS		MIXED	LIGHT	HEAVY		
Nitrogen		---	---	---		---
CO2		---	---	---		---
Methane		---	---	---		---
Ethane		---	---	---		---
Propane		---	---	---		---
i-Butane		---	---	---		---
n-Butane		---	---	---		---
i-Pentane		---	---	---		---
n-Pentane		---	---	---		---
n-Hexane		---	---	---		---
C7PLUS*		---	---	---		---
H2O		---	---	---		---
Material Stream: Outlet (2-phase)					Fluid Package:	Basis-2
					Property Package:	Peng Robinson
CONDITIONS						
	Overall	Vapour Phase	Liquid Phase			
Vapour / Phase Fraction	0.6702	0.6702	0.3298			
Temperature: (C)	108.9	108.9	108.9			
Pressure: (kPa)	9000	9000	9000			
Molar Flow (kgmole/h)	4804	3220	1585			
Mass Flow (kg/h)	2.060e+005	9.127e+004	1.147e+005			
Liquid Volume Flow (m3/h)	395.6	217.5	178.1			
Molar Enthalpy (kJ/kgmole)	-1.167e+005	-9.190e+004	-1.670e+005			
Molar Entropy (kJ/kgmole-C)	160.6	158.8	164.3			
Heat Flow (kJ/h)	-5.605e+008	-2.959e+008	-2.646e+008			
Std Liq Volume Flow (m3/h)	438.0	---	173.8			
PROPERTIES						
	Overall	Vapour Phase	Liquid Phase			
Vapour / Phase Fraction	0.6702	0.6702	0.3298			
Temperature: (C)	108.9	108.9	108.9			
Pressure: (kPa)	9000	9000	9000			

Material Stream: Outlet (2-phase) (continued)				Fluid Package:	Basis-2	
				Property Package:	Peng Robinson	
PROPERTIES						
	Overall	Vapour Phase	Liquid Phase			
Actual Volume Flow (m3/h)	1108	895.9	211.6			
Mass Enthalpy (kJ/kg)	-2722	-3242	-2307			
Mass Entropy (kJ/kg-C)	3.747	5.603	2.270			
Molecular Weight	42.87	28.35	72.37			
Molar Density (kgmole/m3)	4.338	3.594	7.488			
Mass Density (kg/m3)	186.0	101.9	541.9			
Std Liquid Mass Density (kg/m3)	---	---	659.9			
Molar Heat Capacity (kJ/kgmole-C)	115.1	80.00	186.3			
Mass Heat Capacity (kJ/kg-C)	2.684	2.822	2.574			
Thermal Conductivity (W/m-K)	---	4.651e-002	7.495e-002			
Viscosity (cP)	---	1.789e-002	0.1367			
Surface Tension (dyne/cm)	---	---	6.283			
Z Factor	---	0.7884	0.3784			
Molar Vapour Fraction	0.6702	0.6702	0.3298			
Mass Vapour Fraction	0.4431	0.4431	0.5569			
Volume Vapour Fraction	0.5498	0.5498	0.4502			
Molar Volume (m3/kgmole)	0.2305	0.2783	0.1335			
Actual Gas Flow (ACT m3/h)	---	895.9	---			
Actual Liquid Flow (m3/s)	5.878e-002	---	5.878e-002			
Std. Gas Flow (STD m3/h)	1.136e+005	7.613e+004	3.747e+004			
Std. Liquid Volume Flow (m3/h)	438.0	---	173.8			
Watson K	13.91	15.82	12.57			
Kinematic Viscosity (cSt)	---	0.1756	0.2522			
Cp/Cv	1.078	1.437	1.274			
Lower Heating Value (kJ/kgmole)	---	---	---			
Mass Lower Heating Value (kJ/kg)	---	---	---			
Liquid Fraction	0.3298	0.0000	1.000			
Partial Pressure (kPa)	75.34	75.34	75.34			
COMPOSITION						
Overall Phase				Vapour Fraction 0.6702		
COMPONENTS	MOLAR FLOW (kgmole/h)	MOLE FRACTION	MASS FLOW (kg/h)	MASS FRACTION	LIQUID VOLUME FLOW (m3/h)	LIQUID VOLUME FRACTION
Nitrogen	24.7154	0.0051	692.3512	0.0034	0.8586	0.0022
CO2	33.5430	0.0070	1476.2173	0.0072	1.7886	0.0045
Methane	2571.3481	0.5352	41251.8812	0.2003	137.7846	0.3483
Ethane	166.3496	0.0346	5002.1164	0.0243	14.0634	0.0355
Propane	184.2301	0.0383	8123.9945	0.0394	16.0338	0.0405
i-Butane	968.4230	0.2016	56288.6214	0.2733	100.1637	0.2532
n-Butane	28.8233	0.0060	1675.3228	0.0081	2.8725	0.0073
i-Pentane	6.5280	0.0014	471.0019	0.0023	0.7555	0.0019
n-Pentane	413.8960	0.0861	29863.0107	0.1450	47.4220	0.1199
n-Hexane	10.7775	0.0022	928.7829	0.0045	1.4016	0.0035
C7PLUS*	371.5727	0.0773	59748.8853	0.2901	72.0213	0.1821
H2O	24.2833	0.0051	437.4668	0.0021	0.4383	0.0011
Total	4804.4900	1.0000	205959.6524	1.0000	395.6041	1.0000
Vapour Phase				Phase Fraction 0.6702		
COMPONENTS	MOLAR FLOW (kgmole/h)	MOLE FRACTION	MASS FLOW (kg/h)	MASS FRACTION	LIQUID VOLUME FLOW (m3/h)	LIQUID VOLUME FRACTION
Nitrogen	22.5180	0.0070	630.7981	0.0069	0.7823	0.0036
CO2	26.9534	0.0084	1186.2131	0.0130	1.4373	0.0066
Methane	2203.1230	0.6843	35344.4619	0.3872	118.0534	0.5427
Ethane	123.4407	0.0383	3711.8488	0.0407	10.4358	0.0480
Propane	115.6673	0.0359	5100.5827	0.0559	10.0667	0.0463
i-Butane	518.9261	0.1612	30162.0631	0.3305	53.6724	0.2468
n-Butane	14.3612	0.0045	834.7305	0.0091	1.4312	0.0066
i-Pentane	2.6376	0.0008	190.3042	0.0021	0.3052	0.0014
n-Pentane	156.2238	0.0485	11271.7041	0.1235	17.8993	0.0823

Material Stream: Outlet (2-phase) (continued)					Fluid Package:	Basis-2
					Property Package:	Peng Robinson
COMPOSITION						
Vapour Phase (continued)					Phase Fraction	0.6702
COMPONENTS	MOLAR FLOW (kgmole/h)	MOLE FRACTION	MASS FLOW (kg/h)	MASS FRACTION	LIQUID VOLUME FLOW (m3/h)	LIQUID VOLUME FRACTION
n-Hexane	2.9541	0.0009	254.5776	0.0028	0.3842	0.0018
C7PLUS*	13.9364	0.0043	2240.9711	0.0246	2.7013	0.0124
H2O	19.0110	0.0059	342.4846	0.0038	0.3432	0.0016
Total	3219.7527	1.0000	91270.7599	1.0000	217.5123	1.0000
Liquid Phase					Phase Fraction	0.3298
COMPONENTS	MOLAR FLOW (kgmole/h)	MOLE FRACTION	MASS FLOW (kg/h)	MASS FRACTION	LIQUID VOLUME FLOW (m3/h)	LIQUID VOLUME FRACTION
Nitrogen	2.1973	0.0014	61.5531	0.0005	0.0763	0.0004
CO2	6.5896	0.0042	290.0042	0.0025	0.3514	0.0020
Methane	368.2252	0.2324	5907.3993	0.0515	19.7312	0.1108
Ethane	42.9089	0.0271	1290.2676	0.0113	3.6276	0.0204
Propane	68.5628	0.0433	3023.4118	0.0264	5.9671	0.0335
i-Butane	449.4969	0.2836	26126.5583	0.2278	46.4914	0.2611
n-Butane	14.4621	0.0091	840.5922	0.0073	1.4413	0.0081
i-Pentane	3.8904	0.0025	280.6977	0.0024	0.4502	0.0025
n-Pentane	257.6722	0.1626	18591.3066	0.1621	29.5227	0.1658
n-Hexane	7.8234	0.0049	674.2053	0.0059	1.0174	0.0057
C7PLUS*	357.6363	0.2257	57507.9142	0.5014	69.3201	0.3892
H2O	5.2724	0.0033	94.9822	0.0008	0.0952	0.0005
Total	1584.7373	1.0000	114688.8925	1.0000	178.0918	1.0000
K VALUE						
COMPONENTS	MIXED		LIGHT		HEAVY	
Nitrogen	5.044		5.044		---	
CO2	2.013		2.013		---	
Methane	2.945		2.945		---	
Ethane	1.416		1.416		---	
Propane	0.8303		0.8303		---	
i-Butane	0.5682		0.5682		---	
n-Butane	0.4888		0.4888		---	
i-Pentane	0.3337		0.3337		---	
n-Pentane	0.2984		0.2984		---	
n-Hexane	0.1859		0.1859		---	
C7PLUS*	1.918e-002		1.918e-002		---	
H2O	1.775		1.775		---	
Pipe Segment: Well-1						
PARAMETERS						
Pressure Gradient/Pipe Parameters						
2 - Phase Flow Map: ---				Pressure Drop: 1785 kPa		
Length - Elevation Dimensions						
Segment Number	1		2		3	
Fitting/Pipe	Pipe		Pipe		Pipe	
Distance (m)	1000		1000		1000	
Elevation (m)	1000		1000		1000	
Outer Diameter (mm)	203.2		203.2		203.2	
Inner Diameter (mm)	177.8		177.8		177.8	
Material	Mild Steel		Mild Steel		Mild Steel	
Increments	5		5		5	
Specific Pipe Parameters						
Pipe Schedule	Actual		Material		Mild Steel	
Hyprotech Ltd.	HYSYS.Plant v2.2.2 (Build 3806)				Page 4 of 8	

Pipe Segment: Well-1 (continued)				
PARAMETERS				
Length - Elevation Dimensions				
Nominal Diameter	---	Roughness	4.572e-005 m	
Inner Diameter	177.8 mm			
Fittings Parameters				
Fitting Name	Pipe	K Factor	---	Inner Diameter 177.8 mm
Segment Heat Transfer Info				
Segment Number	1	2	3	
Type				
Amb. Temp. (C)	200.0	172.7	145.4	
HTC (kJ/h-m2-C)	2.314	2.314	2.314	
CALCULATION				
Pressure Tolerance (kPa)	0.1000	Length Step Size (m)	1000	
Temperature Tolerance (C)	1.000e-002	Flow Initial Guess (kgmole/h)	360.0	
Heat Flow Tolerance (kJ/h)	0.3600	Flow Step Size (kgmole/h)	252.0	
Length Initial Guess (m)	5000	Default Increments	5	
PROFILES				
Data				
Distance (m)	Elevation (m)		Increments	
0.0000	0.0000		5	
1000	1000		5	
2000	2000		5	
3000	3000			
Pipe Table				
Length (m)	0.0000	200.0	400.0	600.0
Elevation (m)	0.0000	200.0	400.0	600.0
Pressure (kPa)	1.271e+004	1.259e+004	1.246e+004	1.234e+004
Temperature (C)	200.0	199.2	198.3	197.5
Heat Transferred (kJ/h-m)	---	-110.4	-108.6	-107.4
Flow Regime	Vapour Only	Vapour Only	Vapour Only	Vapour Only
Liquid HoldUp	0.0000	0.0000	0.0000	0.0000
Friction Gradient (kPa/m)	1.080e-002	1.088e-002	1.096e-002	1.105e-002
Static Gradient (kPa/m)	0.6172	0.6125	0.6078	0.6031
Accel. Gradient (kPa/m)	0.0000	0.0000	0.0000	0.0000
Liq. Reynolds	---	---	---	---
Vap. Reynolds	1.197e+006	1.200e+006	1.203e+006	1.205e+006
Liquid Velocity (m/s)	---	---	---	---
Vapour Velocity (m/s)	2.001	2.017	2.032	2.048
Length (m)	800.0	1000	1200	1400
Elevation (m)	800.0	1000	1200	1400
Pressure (kPa)	1.222e+004	1.210e+004	1.198e+004	1.186e+004
Temperature (C)	196.7	195.9	194.8	193.8
Heat Transferred (kJ/h-m)	-106.1	-104.9	-143.9	-142.3
Flow Regime	Vapour Only	Vapour Only	Vapour Only	Vapour Only
Liquid HoldUp	0.0000	0.0000	0.0000	0.0000
Friction Gradient (kPa/m)	1.113e-002	1.122e-002	1.130e-002	1.138e-002
Static Gradient (kPa/m)	0.5984	0.5937	0.5895	0.5852
Accel. Gradient (kPa/m)	0.0000	0.0000	0.0000	0.0000
Liq. Reynolds	---	---	---	---
Vap. Reynolds	1.208e+006	1.211e+006	1.214e+006	1.217e+006
Liquid Velocity (m/s)	---	---	---	---
Vapour Velocity (m/s)	2.064	2.080	2.095	2.111
Length (m)	1600	1800	2000	2200
Elevation (m)	1600	1800	2000	2200
Hyprotech Ltd.	HYSYS.Plant v2.2.2 (Build 3806)			Page 5 of 8

Pipe Segment: Well-1 (continued)					
PROFILES					
Pipe Table					
Pressure	(kPa)	1.174e+004	1.162e+004	1.150e+004	1.139e+004
Temperature	(C)	192.7	191.7	190.7	189.4
Heat Transferred	(kJ/h-m)	-140.8	-139.2	-137.7	-176.4
Flow Regime		Vapour Only	Vapour Only	Vapour Only	Vapour Only
Liquid HoldUp		0.0000	0.0000	0.0000	0.0000
Friction Gradient	(kPa/m)	1.146e-002	1.154e-002	1.163e-002	1.171e-002
Static Gradient	(kPa/m)	0.5809	0.5766	0.5724	0.5685
Accel. Gradient	(kPa/m)	0.0000	0.0000	0.0000	0.0000
Liq. Reynolds		---	---	---	---
Vap. Reynolds		1.220e+006	1.223e+006	1.226e+006	1.229e+006
Liquid Velocity	(m/s)	---	---	---	---
Vapour Velocity	(m/s)	2.126	2.142	2.158	2.173
Length	(m)	2400	2600	2800	3000
Elevation	(m)	2400	2600	2800	3000
Pressure	(kPa)	1.127e+004	1.116e+004	1.104e+004	1.093e+004
Temperature	(C)	188.2	186.9	185.7	184.5
Heat Transferred	(kJ/h-m)	-174.5	-172.7	-170.8	-169.0
Flow Regime		Vapour Only	Vapour Only	Vapour Only	Vapour Only
Liquid HoldUp		0.0000	0.0000	0.0000	0.0000
Friction Gradient	(kPa/m)	1.179e-002	1.187e-002	1.195e-002	1.203e-002
Static Gradient	(kPa/m)	0.5646	0.5607	0.5568	0.5528
Accel. Gradient	(kPa/m)	0.0000	0.0000	0.0000	0.0000
Liq. Reynolds		---	---	---	---
Vap. Reynolds		1.233e+006	1.236e+006	1.240e+006	1.243e+006
Liquid Velocity	(m/s)	---	---	---	---
Vapour Velocity	(m/s)	2.188	2.203	2.218	2.234
Pipe Segment: PIPE-101					
PARAMETERS					
Pressure Gradient/Pipe Parameters					
2 - Phase Flow Map: ---			Pressure Drop: 103.7 kPa		
Length - Elevation Dimensions					
Segment Number		1	2	3	
Fitting/Pipe		Pipe	Elbow: 90 Std	Pipe	
Distance	(m)	1500	0.0000	1.000e+004	
Elevation	(m)	0.0000	0.0000	0.0000	
Outer Diameter	(mm)	203.2	---	203.2	
Inner Diameter	(mm)	177.8	177.8	177.8	
Material		Mild Steel	Mild Steel	Mild Steel	
Increments		5	1	5	
Specific Pipe Parameters					
Pipe Schedule		Actual	Material	Mild Steel	
Nominal Diameter		---	Roughness	4.572e-005 m	
Inner Diameter		177.8 mm			
Fittings Parameters					
Fitting Name	Pipe	K Factor	---	Inner Diameter	177.8 mm
Heat Transfer Summary					
Heat Loss:	4.466e+006 kJ/h	Ambient Temp:	5.000 C	Overall HTC:	6.766 kJ/h-m ² -C
Inside Heat Transfer Coefficient Estimation					
Estimate Inner HTC:		Enabled	Correlation:	Petukov	
Outside Heat Transfer Coefficient Estimation					
Hyprotech Ltd.		HYSYS.Plant v2.2.2 (Build 3806)		Page 6 of 8	

Pipe Segment: PIPE-101 (continued)				
PARAMETERS				
Outside Heat Transfer Coefficient Estimation				
Estimate Outer HTC	Outer Diameter	Ambient Medium	Velocity	
Enabled	---	Water	1.000 m/s	
Conduction Heat Transfer Coefficient Estimation				
Include Conduction:	Yes	Type: Urethane Foam	Thermal Cond.: 1.800e-002 W/m-K	Thickness: 1.000e-002 m
CALCULATION				
Pressure Tolerance (kPa)	0.1000	Length Step Size (m)	1000	
Temperature Tolerance (C)	1.000e-002	Flow Initial Guess (kgmole/h)	360.0	
Heat Flow Tolerance (kJ/h)	0.3600	Flow Step Size (kgmole/h)	252.0	
Length Initial Guess (m)	5000	Default Increments	5	
PROFILES				
Data				
Distance (m)	Elevation (m)		Increments	
0.0000	0.0000		5	
1500	0.0000		1	
1500	0.0000		5	
1.150e+004	0.0000			
Pipe Table				
Length (m)	0.0000	300.0	600.0	900.0
Elevation (m)	0.0000	0.0000	0.0000	0.0000
Pressure (kPa)	1.082e+004	1.081e+004	1.081e+004	1.080e+004
Temperature (C)	184.3	177.2	170.3	163.6
Heat Transferred (kJ/h-m)	---	-759.0	-728.7	-699.4
Flow Regime	Vapour Only	Vapour Only	Vapour Only	Vapour Only
Liquid HoldUp	0.0000	0.0000	0.0000	0.0000
Friction Gradient (kPa/m)	1.215e-002	1.192e-002	1.169e-002	1.146e-002
Static Gradient (kPa/m)	0.0000	0.0000	0.0000	0.0000
Accel. Gradient (kPa/m)	0.0000	0.0000	0.0000	0.0000
Liq. Reynolds	---	---	---	---
Vap. Reynolds	1.245e+006	1.255e+006	1.265e+006	1.274e+006
Liquid Velocity (m/s)	---	---	---	---
Vapour Velocity (m/s)	2.256	2.213	2.171	2.130
Length (m)	1200	1500	1500	3500
Elevation (m)	0.0000	0.0000	0.0000	0.0000
Pressure (kPa)	1.080e+004	1.080e+004	1.080e+004	1.078e+004
Temperature (C)	157.2	151.0	151.0	115.6
Heat Transferred (kJ/h-m)	-671.2	-644.1	0.0000	-550.7
Flow Regime	Vapour Only	Vapour Only	Vapour Only	Vapour Only
Liquid HoldUp	0.0000	0.0000	0.0000	0.0000
Friction Gradient (kPa/m)	1.125e-002	1.104e-002	---	9.797e-003
Static Gradient (kPa/m)	0.0000	0.0000	---	0.0000
Accel. Gradient (kPa/m)	0.0000	0.0000	---	0.0000
Liq. Reynolds	---	---	---	---
Vap. Reynolds	1.284e+006	1.293e+006	1.293e+006	1.348e+006
Liquid Velocity (m/s)	---	---	---	---
Vapour Velocity (m/s)	2.091	2.052	2.052	1.825
Length (m)	5500	7500	9500	1.150e+004
Elevation (m)	0.0000	0.0000	0.0000	0.0000
Pressure (kPa)	1.076e+004	1.074e+004	1.073e+004	1.071e+004
Temperature (C)	88.67	68.50	53.46	42.22
Heat Transferred (kJ/h-m)	-416.8	-315.8	-240.3	-184.0
Flow Regime	Vapour Only	Vapour Only	Vapour Only	Vapour Only
Liquid HoldUp	0.0000	0.0000	0.0000	0.0000
Friction Gradient (kPa/m)	8.811e-003	8.036e-003	7.433e-003	6.965e-003
Static Gradient (kPa/m)	0.0000	0.0000	0.0000	0.0000

Hyprotech Ltd.

HYSYS Plant v2.2.2 (Build 3806)

Page 7 of 8

Pipe Segment: PIPE-101 (continued)				
PROFILES				
Pipe Table				
Accel. Gradient (kPa/m)	0.0000	0.0000	0.0000	0.0000
Liq. Reynolds	---	---	---	---
Vap. Reynolds	1.389e+006	1.415e+006	1.429e+006	1.436e+006
Liquid Velocity (m/s)	---	---	---	---
Vapour Velocity (m/s)	1.643	1.500	1.388	1.300
Adjust: ADJ-5				
Adjusted Variable		Measured Variable		
OBJECT	VARIABLE	OBJECT	VARIABLE	
VLV-101	Pressure Drop	Outlet (2-phase)	Pressure	
Solving Parameters				
Source for Target Value:	User Supplied	Target Value:	9000 kPa	
Solving Method:	---	Tolerance:	1.000 kPa	Maximum Iterations: 200
Step Size:	1000 kPa	Maximum:	---	Minimum: ---
Iteration History				
Iteration Number	Adjust Value		Target Value	
Hyprotech Ltd. HYSYS.Plant v2.2.2 (Build 3806) Page 8 of 8				

