

# The Engineering of Petroleum Streams

by

Mohammad Faizul Hoda

A dissertation submitted in partial  
fulfillment of the requirements for  
the degree of

Doktor Ingeniør

Department of Petroleum  
Engineering and Applied  
Geophysics, Norwegian University  
of Science and Technology

June, 2002



## ABSTRACT

Department of Petroleum Engineering and Applied Geophysics

Norwegian University of Science and Technology

### **THE ENGINEERING OF PETROLEUM STREAMS**

by

Mohammad Faizul Hoda

In a multidisciplinary petroleum industry using a wide variety of engineering software, consistent and easy transfer of quantitative data related to petroleum streams is a major issue. Making it more difficult is the lack of cross-disciplinary understanding of the source and utility of streams data. This may lead to misuse of the streams data. This thesis attempts to address the key issues of petroleum streams management and how multi-disciplinary engineering can use and exchange streams data in a concise and accurate manner. Some of the research in this thesis has led to developments in new state-of-the-art software, Streamz, and all of the applications studied in this thesis have benefited from the Streamz software authored by Dr. Aaron A. Zick.



## ACKNOWLEDGMENTS

I wish to express my sincere gratitude to my supervisor Professor Curtis H. Whitson for providing me with excellent academic, professional and personal guidance during my 6 years at the Institute of Petroleum Technology. This thesis is the fruit of his original concepts and profound experience. The immensely interesting topic covered in this thesis has seldom been addressed in the open literature and working on it with him has been both challenging and fun.

Professor Michael Golan was my other supervisor. I would like to thank him for all the help and the fruitful discussions we had together. I am indebted to him for introducing me to Curtis as a possible dr.ing. candidate. Discussions with Mike always broadened the perspective and he always gave insight from a new point of view.

The methods presented in this thesis revolve around the development of Streamz by Dr. Aaron A. Zick. This program was used exclusively for most of the testing and field examples presented. Aaron's presence and the frequent interactions with him instilled the value of accuracy, thoroughness and minute detail. He became a "virtual" advisor and I sincerely thank him for all his help.

Friends and colleagues at Pera a/s and at IPT provided opportunities for many interesting discussions that have contributed to this work. Thanks to Kameshwar Singh, Øivind Fevang, Vidar Haugse, and Tao Yang. Special mention is due to Knut Uleberg with whom I worked many hours, weeks and months on field applications. Kathy at Pera and secretaries at IPT were always keen to help whenever needed.

I wish to acknowledge the contributions of the Norwegian Research Council, Norsk Hydro a/s and Pera a/s in providing financial support during the work.

At the end I wish to dedicate this thesis to my father who passed away recently. He and my mother were a source of constant encouragement. I am indebted to my wife Asfa, and children Rafid, Mushir and Zaina, for suffering all these years. This is as much their accomplishment as mine and I look forward to making up.

Trondheim, June 2002  
Faiz ul Hoda



## TABLE OF CONTENTS

<b>Abstract</b> .....	<b>i</b>
<b>Acknowledgments</b> .....	<b>iii</b>
<b>Concepts, Issues and Background</b> .....	<b>1</b>
1.1 Introduction .....	1
1.2 Conversions .....	1
1.3 Management.....	3
1.4 Processing.....	4
1.5 Organization and Background .....	5
<b>Stream Conversion Methods</b> .....	<b>7</b>
2.1 Concept of the Generic Conversion.....	7
2.2 Gamma Distribution Conversions.....	9
2.2.1 The Gamma Distribution Model.....	9
2.2.2 The Truncated Gamma Distribution Model.....	11
2.2.3 Use of Gamma Distribution for Petroleum Fluids.....	12
2.2.4 From the Gamma Model to Discrete Fractions .....	18
2.2.5 TBP Stream Data Fit.....	19
2.2.6 Splitting of a Single C <sub>7+</sub> Fraction .....	20
2.2.7 Pseudoization of EOS Characterization.....	21
2.2.8 Comparison of Exponential vs. Gamma Fit.....	21
2.2.9 Typical Gamma Fit usage for Multiple Samples.....	22
2.2.10 Gamma Distribution Application to Reservoir Streams .....	24
2.2.11 Gamma Distribution Formulation as Split Factors .....	33
2.2.12 Conclusions .....	33
2.3 BOz conversion .....	34
2.3.1 Split Factors for Black-oil to Compositional (BOz) Conversions .....	37
2.3.2 Negative Split Factors?.....	42
2.3.3 Implementation of Gas Injection .....	47
2.3.4 Error in Oil Rates.....	48
2.3.5 Why BOz Conversions are Necessary on a Connection Basis? .....	54
2.3.6 Quality Checks on BOz results .....	58
2.3.7 Negative Molar Rates .....	60
2.3.8 BOz Conversion Examples .....	68
2.3.9 Conclusions .....	76
2.4 Thermodynamic Conversions .....	77
2.4.1 Split Factor based Well Test Conversions .....	77
2.4.2 Streamz Process command based Well Test Conversions .....	82
2.4.3 Delumping.....	85
2.4.4 Conclusions .....	92
<b>Streamz Command Syntax &amp; Examples</b> .....	<b>93</b>
3.1 Introduction .....	93

3.2 Streamz Concepts.....	93
3.2.1 Characterization .....	93
3.2.2 Stream .....	94
3.2.3 Conversion.....	95
3.3 Streamz Commands.....	95
3.3.1 General Program and I/O Control Commands.....	95
3.3.2 Fluid Characterization Commands .....	97
3.3.3 Characterization Conversion Commands .....	99
3.3.4 File Control Commands.....	99
3.3.5 Variable and Domain definition Commands.....	100
3.3.6 Stream Manipulation and Output Commands.....	101
3.3.7 Stream Processing Commands .....	103
3.4 Streamz Commands used in Examples .....	104
3.4.1 North Sea Full-Field Case.....	104
3.4.2 HTHP Rich Gas Condensate Reservoir: Pressure Depletion.....	111
3.4.3 Rich Gas Condensate with Compositional Gradient.....	117
3.4.4 Gas Injection in Rich GC reservoir with Constant Composition...	123
3.4.5 TBP Stream Data Fit .....	125
3.4.6 Splitting of a Single C <sub>7+</sub> Fraction.....	127
3.4.7 Typical Gamma Fit usage for Multiple Samples .....	129
3.4.8 Gamma Distribution Application to Reservoir Streams .....	131
3.4.9 Streamz Process command based Well Test Conversions .....	134
3.4.10 Conclusions .....	139
<b>References.....</b>	<b>141</b>
<b>Nomenclature .....</b>	<b>143</b>
<b>Appendix A: Typical Test and Reference Separator Ratios.....</b>	<b>149</b>
<b>Appendix B: File Format for Extended Black-Oil PVT Data.....</b>	<b>155</b>
<b>Appendix C1: North Sea Full Field.....</b>	<b>165</b>
<b>Appendix C2: CO<sub>2</sub> Immiscible Injection.....</b>	<b>175</b>
<b>Appendix C3: HTHP Rich GC Reservoir: Pressure depletion .....</b>	<b>183</b>
<b>Appendix C4: Rich GC with Compositional Gradient .....</b>	<b>193</b>
<b>Appendix C5: Rich GC Reservoir with Constant Composition .....</b>	<b>203</b>
<b>Appendix C6: TBP stream data Fit.....</b>	<b>211</b>
<b>Appendix C7: Splitting of a Single C<sub>7+</sub> fraction. ....</b>	<b>215</b>
<b>Appendix C8: Comparison of Exponential vs. Gamma Fit .....</b>	<b>219</b>
<b>Appendix C9: Typical Gamma Fit usage for Multiple Samples .....</b>	<b>225</b>
<b>Appendix C10: Gamma Distribution application to Reservoir Streams</b>	<b>233</b>

# Chapter 1

## Concepts, Issues and Background

### 1.1 Introduction

The petroleum industry today is highly dependent on technology. Computer technology is an integral part of the drive to optimize and maximize recovery of petroleum fluids from reservoirs. A multitude of computer techniques, models and applications are used by the many engineering disciplines in any petroleum company. Each one of them uses a representation of **petroleum fluids streams** in some form.

Use of various off-the-shelf, or custom designed, software packages in the petroleum industry is very widespread. This may lead to a basic mismatch and inconsistency in the information exchanged among these programs. The quantitative description of petroleum streams (e.g. gas, oil, and water) as required by different engineering disciplines - reservoir, production, process, pipeline, and refining - is handled very differently even by similar software dealing with similar solutions. Stream descriptions can differ considerably in each engineering discipline, and sometimes even within the same discipline e.g. black-oil and compositional reservoir simulation models.

The handling of petroleum fluid streams in a consistent, easy and documented manner is a major goal that is not easily perceived or appreciated. This thesis discusses the **engineering of petroleum streams** in this context. Various technologies are developed and tested for consistent **conversions** among the various quantitative representations of petroleum fluid streams. Technologies developed for **management** of huge petroleum fluid data are tested and documented. Much of the consistent handling of petroleum fluids involves the ability to **process** streams in an easy and consistent manner.

### 1.2 Conversions

Conversion among various representations of petroleum fluid streams is the major problem investigated by this dr.ing. work. There are various issues related to stream conversions. The major and underlying issue is **consistency**. When petroleum fluid data is transferred from one engineering discipline to

another a conversion is usually required. This conversion should have a physical and logical basis for the resulting stream to be consistent with the original. User defined **constraints** (e.g. conservation of mass or moles) should be enforced.

Reservoir simulation is usually performed using the most efficient model possible. This frequently requires the use of black-oil models. Coats and Fevang et. al.<sup>1</sup> have developed concrete guidelines for determining when a black-oil model is sufficient and when a compositional model is necessary. The results of such black-oil simulations are frequently fed to a sophisticated process simulator to determine the “revenue” product profiles. These process simulators need a detailed compositional representation. How is this conversion done? The reservoir engineers usually consider their work done when their simulation runs as fast as possible and is consistent with their expected results. The process engineers get a black-oil profile of surface gas and surface oil rates (probably very detailed with every simulator time step).

How does the process engineer use this data? They probably use some black-box method, not known to the reservoir engineers, to generate a detailed compositional stream for their process calculation. Some methods start with a sample composition (e.g. initial production test sample, or original DST sample). This is usually scaled to plant design capacities to obtain molar rates. Some amount of 1<sup>st</sup> stage separator gas is added to obtain the producing GOR. Usually no consideration is given to the separation specified in the reservoir simulator; No consideration is given to actual oil and gas rates resulting in the GOR. No consideration is given to variation of composition with time.

An accurate, consistent, easy to use, and documented method is needed for this type of frequently required **black-oil to compositional** (termed **BOz** in this thesis) conversion. The single most important achievement of this thesis is the development and testing of an accurate BOz method.

Even when the fluid system requires a compositional model, 5-8 components are used at the most. Various authors<sup>1,2</sup> have described procedures for this “lumping” of components. The results of such simulations frequently need to be converted to more detailed representations. While lumping detailed components into pseudo-components the conversion is trivial. The inverse is non-trivial. Authors<sup>3,4</sup> have investigated this inverse lumping or de-lumping and most map or track the entire process that results in the compositions in the producing well connections. Simpler methods based on the **Gamma Distribution** for the heptanes plus fractions are investigated and applied. For the heptanes minus fractions compositional and/or pressure dependent conversions are proposed for this **compositional to compositional** conversion.

Many **applications** that handle petroleum fluid streams use different and proprietary **formats**. This frequently requires users to go through still other (non-petroleum) applications to transfer data. The manual work required is an invitation to errors. The technologies developed for conversion of petroleum fluid streams should transparently read the source formats and write to destination formats.

There is a need for a clearly defined solution to manage the conversion of petroleum fluid streams using clearly **documented** technology that ensures the **accuracy** of data being handled and transferred among unlike formats and applications. The accuracy of conversions is an important issue. Frequently, ad-hoc and approximate solutions are used that might be good enough in certain situations. But these solutions are frequently extended without testing, sometimes resulting in inaccurate data being used. The efficient and fast implementation of the method used by BOz and GD allow the conversions at each **grid-connection** for each **time step**. This ensures improved accuracy.

The **limitations** of the conversion method and the quantification of the potential errors are also addressed.

### 1.3 Management

A reservoir simulator produces thousands, if not millions, of streams. A typical field with sufficiently different geology and fluid systems would normally require separate fluid models for each identified reservoir unit. This translates into **huge databases** of petroleum fluid streams. With the application of state-of-the-art technology, **real-time data acquisition** systems generate huge amounts of production data.

There is a real need for **filtering** streams data. What is the production from a particular region? What is the molar rate of methane from Well ABC between year 2002 and 2003? What is the layer contribution in a non-communicating multi-layer system? To answer such questions an efficient filtering technique is required.

What is the yearly average production rate entering a particular platform? What are the average 22 component profiles? Does the peak field rate exceed the design capacity? These questions need accurate **integration** of rates over any time period. **Aggregation** on the basis of **time** and a common **characterization** is required before mixed streams can be **added**. At what level should the streams be converted? What should be the common denominator representation?

**Multiple fields** frequently flow into common platforms. Need arises to back allocate the production to fields, pipelines, reservoirs, wells and regions in a model. Frequently properties are linked to **sales contracts** and accurate accounting of the sources is needed.

## 1.4 Processing

A reservoir simulation needs hours, days or weeks to run. Some sort of **surface processing** is built into the model. The surface volumes it reports reflect this surface processing. To perform sensitivities on changing surface processing a simulation is frequently re-run multiple times. Is there a way to avoid this? Can we not **re-process** the produced streams through any other process? Does one require a sophisticated process simulator? Or does the surface process **evaluation** necessitate re-running the simulation?

Production wells undergo monthly **well tests** by diverting them from the main process to a **test separator**, and accurately recording the gas and oil rates. How does this petroleum fluid stream data relate to the actual process? Is there an accurate and consistent method to convert the measurements to **process rates**?

Production tests are conducted by flowing a well through a test separator. The conditions of the separator may vary abruptly and/or smoothly over time. How does this affect the recorded rates? To do a consistent analysis based on this test, the reported rates have to be corrected to a common (un-changing) separator condition. What is the most consistent procedure to do this? Is there an automated way to achieve well test **rate correction**?

Production from a compositional reservoir simulator reports total surface oil and gas rates. Many times the **source allocation** is required to determine the efficiency of miscible production mechanisms. How much of the oil is from free reservoir oil and how much is the **condensate** from reservoir gas? How much of the total gas is **solution-gas**?

An important activity performed to bring the results of reservoir simulators closer to reality is **history matching**. The model is tuned to physically recorded oil and gas rates and pressures by altering geology, rock properties and fluid properties. Monthly well test data provide the rates that are used. But is this consistent? Should not the well test rates be adjusted to the process used in the simulator? What is the most consistent, accurate, and simple way of doing this?

With **multiple partners** owning the production from a field, **allocation** procedures need to be accurate and clearly documented. Frequently allocation is done based on monthly testing of individual wells. But the production should actually be re-calculated based on the actual process. Partnership contracts are sometimes based on different processing. The correct processing needs to be used. Can partner allocation be done with on-line multi-phase metering?

## 1.5 Organization and Background

The research presented in this thesis is based around the development and application of the Streamz software. The development of this software was undertaken to help answer the questions raised in this chapter. Much of the research was the use of this software to solve problems related to these issues. The software was used at each stage of its development and problems, inconsistencies and bugs reported to the author Aaron A. Zick.

Hence much of the discussion will revolve around description of solutions provided by Streamz to the issues of **conversion**, **management**, and **processing** discussed in this chapter. References to Streamz and its commands occur throughout the thesis.

Two of the conversion technologies implemented in Streamz are **BOz** and **Gamma Distribution**. They provide the bread and butter conversions and address most of the issues related to it. The theory behind these technologies is described in Chapter 2. Description of their adaptation for implementation in Streamz is also discussed.

Chapter 2 also discusses the thermodynamic conversions implemented during the research, and a couple of approaches previously published. Numerous field examples of the conversion technologies are presented at the end of Chapter 2.

Application of Streamz to field examples also highlights the requirement of streams **management** capability. The ability and versatility of Streamz to handle tens of thousands of streams, filter, combine and aggregate them is shown by a detailed discussion of Streamz data sets in Chapter 3. Selected examples from Chapter 2 are dissected line-by-line showing the command usage and the reasoning behind them.

The unique **process** command of Streamz is discussed in detail in Chapter 3. This allows the processing of any stream by either an EOS flash, K-value lookup, or generic split factor lookup. This results in easy implementation of

automated **well test correction, surface re-processing, source allocation, history matching** calculations, and automated partner **back-allocation** solutions.

Reference is also frequently made to the **Petrostream Management** (PSM) software. This is a suite of programs with Streamz as its “engine” that allows the linking of Streamz to other applications. One of its modules allows the automatic generation of the BOz split factors.

## Chapter 2

### Stream Conversion Methods

#### 2.1 Concept of the Generic Conversion

**Conversion** between petroleum fluid streams represented in different ways is the basic problem investigated by this dr. ing. thesis. The different forms of representation are called **characterizations**. A characterization defines the number and names of components in a stream. Each stream will contain some quantities of each component. The quantities may have units of volume, mass, moles or some generic amount.

In the most general form, a stream conversion from one characterization to another assumes that a **portion** of the quantity of each component of the source stream ends up (or partitions into) one or more of the components of the destination stream. Coats<sup>2</sup> discusses such a formulation in his idea of pseudoization. He stresses that pseudoization can be either a lumping of components or a combining of streams. Although he limits his discussions to mixtures of  $n$  components dividing into mixtures of  $m$  components with  $m < n$ , the idea and the equations are essentially the same. This portion, called the **split factor**, typically ranges from 0 to 1 (although special conversions may result in negative split factors). For molar and mass-based streams the split factors also would typically sum to 1. For volumetric and amount-based streams and for other non-typical conversions these restrictions may not apply.

Irrespective of the restriction on the range and sum of split factors, the generic conversion concept says that for any component of the destination stream, its quantity value is the **weighted sum** of quantity values of all the components of the source stream. The weights are the portions (i.e. the split factors) of each source component that ends up in that particular destination component. If  $u_i$  is the particular destination (or output) component quantity and  $v_1$  to  $v_m$  are the quantities of each of the source (or input) components, then:

$$u_i = S_{i1}v_1 + S_{i2}v_2 + \dots + S_{ij}v_j + \dots + S_{im}v_m \dots\dots\dots (2.1)$$

Here it is assumed that there are  $m$  components in the source stream. Each of  $S_{ij}$  is the portion of the  $j^{\text{th}}$  source component that ends up in the  $i^{\text{th}}$  destination

component. In other words,  $S_{ij}$  is the split factor of the  $j^{\text{th}}$  source component for the  $i^{\text{th}}$  destination component. This is expressed as:

$$u_i = \sum_{j=1}^m S_{ij} v_j \dots\dots\dots(2.2)$$

This concept of the generic conversion allows any value for the split factors. With typical petroleum streams containing mass or molar quantities, and for conversions from mass-to-mass or moles-to-moles, the range of  $S_{ij}$  will be the typical 0 to 1. This implies that mass or molar quantity is not destroyed nor can it exceed that originally present (conservation of matter). For typical conversions of such stream to similar quantities (i.e. mass to mass or moles to moles) the sum of  $S_{ij}$  will also be 1.

For special applications, like the BOz method discussed later, change in units from volume to moles necessitates split factors that are neither in the range of 0 to 1, nor is the constraint of their sum being unity applicable. Under certain circumstances the split factors for one of the component can also be negative, with the split factor and the volume quantity of the other component compensating for the negative value.

The generic formulation of the conversion matrix allows non-typical conversions limited only by the imagination. For example a simple conversion might be the addition of the amount of each of the  $C_{7+}$  fractions of a stream resulting in a destination stream with a single component (the  $C_{7+}$  amount). If  $z$  is the single output component and there are 7 components in the source stream where the 3 last ones make up the  $C_{7+}$ , then,

$$z = \sum_{j=1}^7 S_j v_j \dots\dots\dots(2.3)$$

$$z = S_1 v_1 + S_2 v_2 + S_3 v_3 + S_4 v_4 + S_5 v_5 + S_6 v_6 + S_7 v_7 \dots\dots\dots(2.4)$$

$$z = 0 \cdot v_1 + 0 \cdot v_2 + 0 \cdot v_3 + 0 \cdot v_4 + 1 \cdot v_5 + 1 \cdot v_6 + 1 \cdot v_7 \dots\dots\dots(2.5)$$

$$z = v_5 + v_6 + v_7 \dots\dots\dots(2.6)$$

Other non-typical conversions can be set up using this generic split factor matrix formulation.

## 2.2 Gamma Distribution Conversions

Conversions of petroleum streams can use the Gamma Distribution (GD) model. However the GD model is only applicable to the heavy fractions (typically C<sub>7+</sub>). Hence this is usually applied in combination to some other method for the lighter components.

### 2.2.1 The Gamma Distribution Model

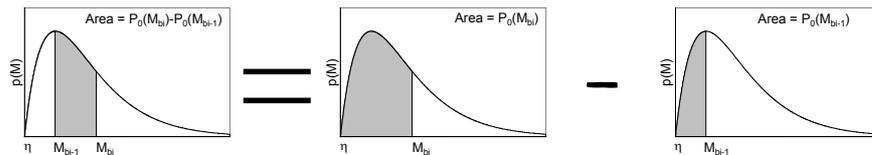
Efforts<sup>5,6,7</sup> to mathematically model the molar distribution of heavy fractions have shown that the exponential distribution is sufficient in most cases. But this assumes that all reservoir fluids having a given C<sub>7+</sub> molecular weight have the same molar distribution, which is realistically not the case. The three-parameter gamma distribution is a more general model. The Gamma distribution model is based on the Gamma Distribution function, which is a skewed (as opposed to normal) probability density function. It relates the frequency of occurrence (normalized mole fraction) to the molecular weight. It is a generalized function that reduces to many known functions for particular values of its parameters. The distribution is characterized by three parameters that determined its shape, size and origin.

The gamma probability density function is

$$p(M) = \frac{(M - \eta)^{\alpha-1} \exp\{-[(M - \eta)/\beta]\}}{\beta^\alpha \Gamma(\alpha)} \dots\dots\dots (2.7)$$

where  $\Gamma$ =gamma function,  $\eta$  is origin of the distribution,  $\alpha$  determines the shape and  $\beta$  is given by

$$\beta = \frac{M_{7+} - \eta}{\alpha} \dots\dots\dots (2.8)$$



**Fig. 2.1 – Mole fraction obtained by differencing of area under the probability density curve within adjacent molecular weight bounds.**

For specified values of M<sub>7+</sub>,  $\alpha$ , and  $\eta$  Eq. (2.7) represents a unique relationship between normalized mole fraction and molecular weight of the C<sub>7+</sub> fraction. This curve is defined from a lower limiting value molecular weight =  $\eta$  to an

upper value of infinity. The total area under the curve is, by definition, unity. The continuous distribution  $p(M)$  is applied to petroleum fractions by dividing the area under the curve, i.e the integral of  $p(M)$  (say  $P_0(M)$ ) into sections as shown in Fig. 2.1. The area of a section is the normalized mole fraction  $z_i/z_{7+}$  for the range of molecular weights between the boundaries  $M_{bi-1}$  to  $M_{bi}$ . This area is obtained by integrating the function at those boundaries, and then taking the difference.

$$\frac{z_i}{z_{7+}} = \int_{M_{bi-1}}^{M_{bi}} p(M)dM = P_0(M_{bi}) - P_0(M_{bi-1}) \dots\dots\dots(2.9)$$

Average molecular weight  $M_i$  of this pseudo-component is given by

$$M_i = \frac{\int_{M_{bi-1}}^{M_{bi}} Mp(M)dM}{z_i / z_{7+}} = \eta + \alpha\beta \frac{P_1(M_{bi}) - P_1(M_{bi-1})}{P_0(M_{bi}) - P_0(M_{bi-1})} \dots\dots\dots(2.10)$$

The cumulative probability function  $P_0(M_{bi})$  can be expressed as an infinite series (in which form it is implemented in computer programs)

$$P_0(M_{bi}) = \frac{e^{-y} y^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{\infty} \frac{y^j}{\prod_{k=0}^j (\alpha + k)} \dots\dots\dots(2.11)$$

as a function of  $\alpha$  and  $y=(M_b-\eta)/\beta$ . Similarly the  $P_1(M_{bi})$  function is expressed as

$$P_1(M_{bi}) = \frac{e^{-y} y^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{\infty} \frac{y^j}{\prod_{k=0}^j (\alpha + k)} - \frac{e^{-y} y^\alpha}{\alpha\Gamma(\alpha)} \dots\dots\dots(2.12)$$

Frequently, the terms in Eqs. (2.11) and (2.12) are broken up

$$\frac{e^{-y} y^\alpha}{\Gamma(\alpha)} = Q \dots\dots\dots(2.13)$$

$$\sum_{j=0}^{\infty} \frac{y^j}{\prod_{k=0}^j (\alpha + k)} = S \dots\dots\dots(2.14)$$

So that

$$P_0(M_{bi}) = QS \dots\dots\dots (2.15)$$

$$P_1(M_{bi}) = QS - \frac{Q}{\alpha} = P_0(M_{bi}) - \frac{Q}{\alpha} \dots\dots\dots (2.16)$$

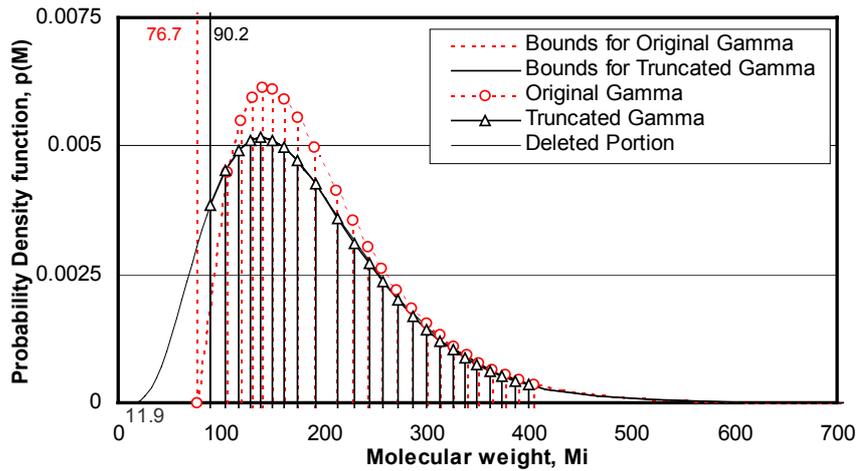
The implementation in Streamz is slightly different where  $P_1$  is calculated first as

$$P_1(M_{bi}) = Q \left( s - \frac{1}{\alpha} \right) = Q \sum_{i=1}^{\infty} \frac{y^i}{\prod_{k=0}^i (\alpha + k)} \dots\dots\dots (2.17)$$

$P_0$  is then obtained as

$$P_0(M_{bi}) = P_1(M_{bi}) + \frac{Q}{\alpha} \dots\dots\dots (2.18)$$

**2.2.2 The Truncated Gamma Distribution Model**



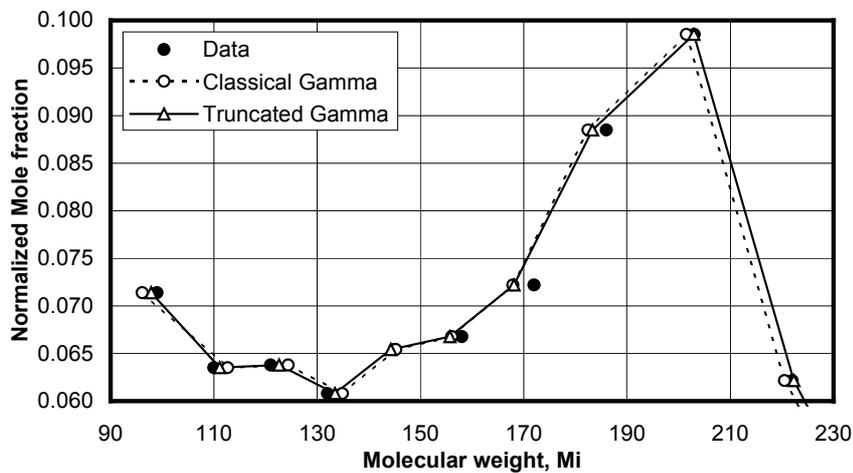
**Fig. 2.2 – An example fluid stream data fit using the Original and the Truncated Gamma distribution. Truncating the GD is essentially the elimination of a portion of the full GD at the lower bound of the lowest MW of the data points.**

A **truncated** GD model has been implemented in Streamz. Zick<sup>8</sup> has observed that many times petroleum fluid data fits the GD very well but

results in a rather low value of MW at which the function goes to zero. Actually there is a gap between the MW at zero of the GD and the lowest MW of the data points. His solution was to introduce a fourth parameter that specifies this lowest MW, and to essentially eliminate the extra portion, truncating the GD at that point. This is seen to give better fit to some petroleum fluids data. In this thesis the original GD is referred to as the Classical Gamma while the 4 parameter GD is labeled the Truncated Gamma.

Fig. 2.2 compares the fit of the Original GD model (circles) with that of the new Truncated model (triangles). The MW at which the Classical GD is zero equals 76.7 where as the lowest MW of the data equals 99. The lower bound of the Truncated GD is calculated as 90.2, which is more consistent with the reported data. The value at which the truncated GD goes to zero is 11.9. The GD below 90.2 (shown with a thin line) is deleted.

Fig. 2.3 compares the corresponding molar distribution where the better fit to measured data is evident.



**Fig. 2.3 – Molar distribution of an example stream comparing the slightly better fit of the Truncated Gamma distribution over the Classical Gamma model. A zoomed portion has been selected.**

### 2.2.3 Use of Gamma Distribution for Petroleum Fluids

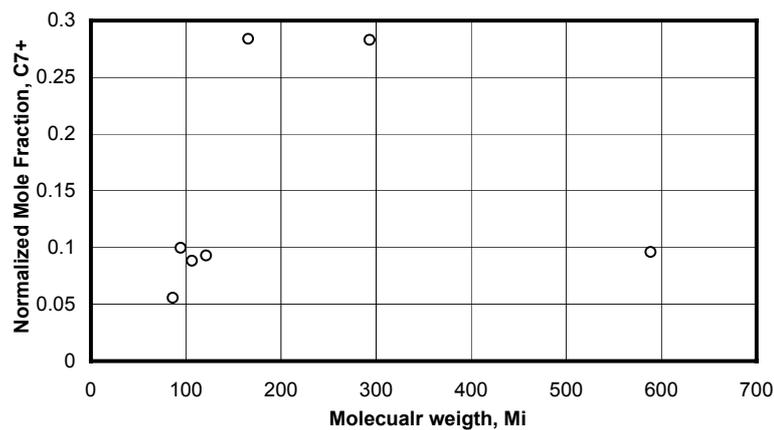
Gamma distribution has been shown by Whitson et.al.<sup>5</sup> to characterize quite well the molar distribution of  $C_{7+}$  fractions of many different fluid systems. They have shown that applying the Gamma distribution allows the varying fluids to be treated with the same EOS model with identical critical properties. Only the amount of each fraction varies as the type of fluid varies.

While converting petroleum streams from one representation to another, Gamma distribution thus offers a very valid and tested procedure. For example, a reservoir stream calculated by a reservoir simulator may have 7  $C_{7+}$  components. When transferring the stream to a Process simulator representation, the stream needs to be represented by 14  $C_{7+}$  components. This conversion needs to be done for each stream, the composition of which may vary with time and also depend on:

- Where in the reservoir it is flowing.
- How it is mixing with other streams.

Converting from one to the other based on a constant and simple split of components is extremely naïve. But using the Gamma distribution fitting and splitting ensures that individual attention is paid to each and every stream on which the conversion is performed.

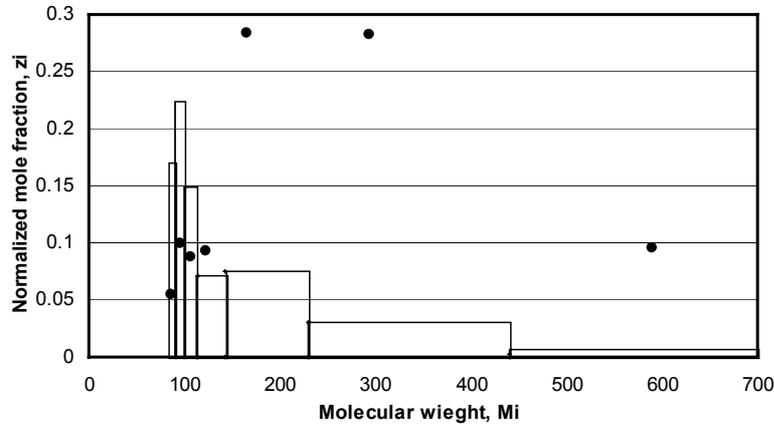
For Gamma distribution fitting, the molar fractions (or mass fractions if chosen so) of each stream, and their molecular weights, are the input data. A plot of the normalized mole fraction vs. molecular weight of an example stream is shown in Fig. 2.4.



**Fig. 2.4 – Molar distribution of an example stream**

The normalized mole fractions are represented as points on the chart at specific molecular weight values. Since each  $C_{7+}$  fraction is actually a grouping of many pure components, these specified molecular weights are actually averages. They are averages of a range of single carbon number components with fixed molecular weights. So the points actually represent the mole fraction of components within molecular weight **bounds** and not fixed molecular weights.

If the normalized mole fractions were now represented as adjacent areas occupied by rectangles with width equal to the width of the molecular weight bounds, a corresponding plot would resemble that shown in Fig. 2.5. This figure is an accurate to-the-scale transformation of Fig. 2.4 with the areas proportional to the mole fractions.



**Fig. 2.5 – Molar distribution of an example stream represented as adjacent rectangles with the areas of rectangles equaling the values of mole fractions.**

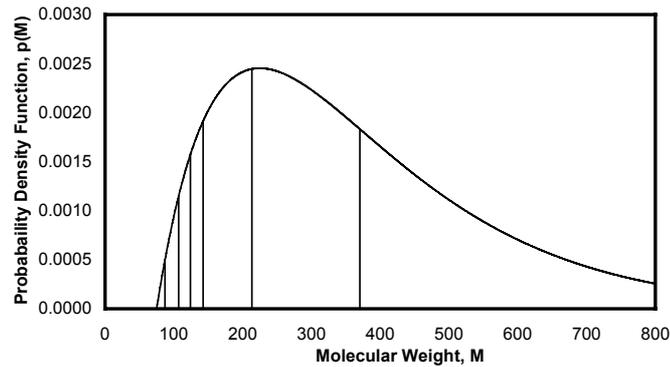
Here the bounds are arbitrarily fixed at arithmetic averages of each fraction molecular weight. The lower bound of the first fraction is also arbitrarily fixed at 84. Any other choices for the bounds could be used and would merely alter the width and height of the rectangles, while keeping the area the same. Fig. 2.4 and Fig. 2.5 are different ways to represent the molar distribution of the same stream.

Fitting the Gamma distribution to measured mole fractions and molecular weights is another way of representing the molar distribution in the form of adjacent areas. The difference is in the manner of choosing a) the initial lower bounding molecular weight, b) the lower bounds of all the fractions (i.e. the widths of the rectangles), and c) the relative heights of rectangles. The choice is performed by regression to obtain the best values of the Gamma distribution parameters:

- **Alpha (or Shape):** Determines the shape of the Gamma distribution, contributing to the height and width of the rectangles. This parameter is also represented by  $\alpha$  in equations and discussions.
- **Eta (or Bound):** Determines the starting point of the truncated Gamma distribution, effectively determining the initial lower

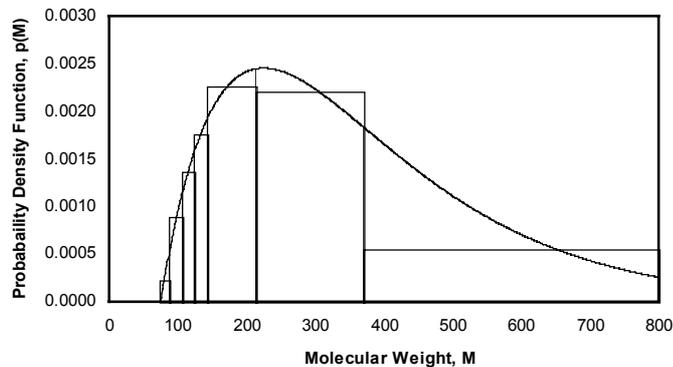
bounding molecular weight. This parameter is also represented by  $\eta$  in equations and discussions.

- **Ave:** The average molecular weight of the fractions being modeled. Determines the size of the Gamma distribution, contributing to the height and width of the rectangles.
- **Origin:** Determines the point at which the Gamma distribution is zero. This parameter is also represented by  $\eta^o$  in discussions.



**Fig. 2.6 – Gamma Molar distribution of an example stream represented as adjacent areas.**

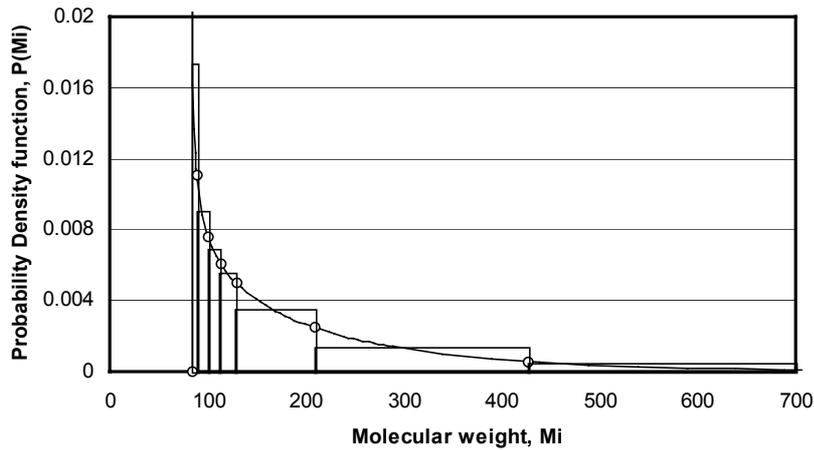
In association with the parameters and input data, and also the material to be conserved, the lower bounding molecular weight of each fraction is determined. Fig. 2.6 shows the Gamma distribution function fit to the stream. The value of Alpha (shape) was fixed at 2 to obtain the **classical** shape of the Gamma distribution. Unrestrained regression yields a shape of 0.75012.



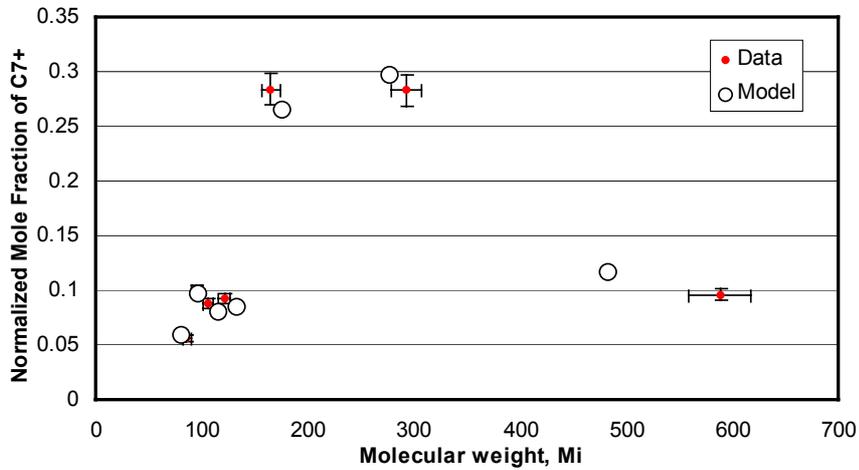
**Fig. 2.7 – Molar distribution of an example stream represented as adjacent rectangles.**

The Gamma molar distribution is a continuous distribution. The area under the probability density function is directly correlated to the normalized mole fraction. Hence we obtain a representation in terms of adjacent areas.

The areas under the function within each molecular weight bound can now be approximated by adjacent rectangles. The width of each rectangle would equal the bounds calculated during the “fit”. The height would be fixed such that the area equals the actual area under the curve for that fraction. This approximation is shown in Fig. 2.7.

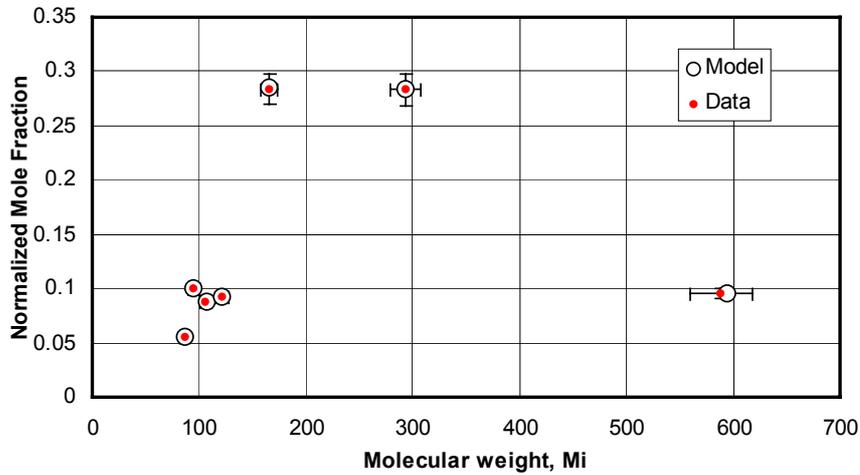


**Fig. 2.8 – Unrestrained Gamma distribution fit of example stream. This is closer looking to the manually generated Fig. 2.5.**

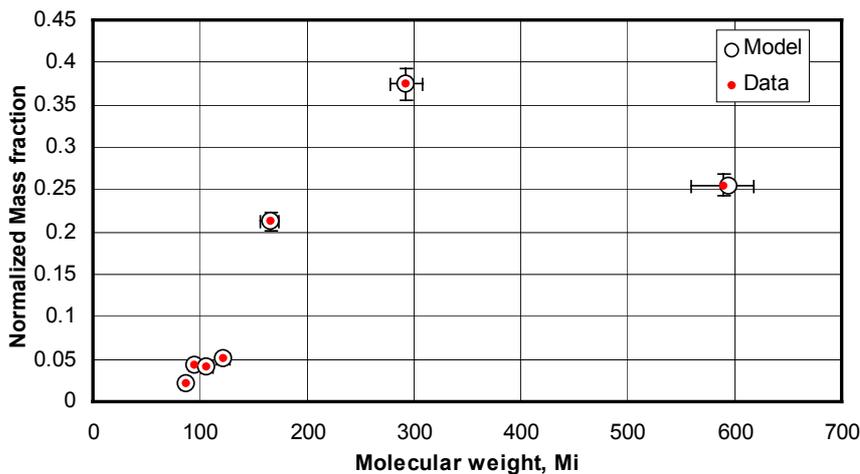


**Fig. 2.9 – Comparing measured and model Molar distributions (Shape restricted to a value of 2).**

Fig. 2.5 and Fig. 2.7 are directly comparable. While Fig. 2.5 is the raw data represented as adjacent rectangles, Fig. 2.7 represents the corresponding GD model as adjacent rectangles. The difference in appearance is due to enforcing the Shape = 2 constraint. Removing this constraint gives a closer looking, but not the typical, Gamma molar distribution of Fig. 2.8.



**Fig. 2.10 – Comparing measured and model Molar distributions (Unrestricted fit yielding Shape of 0.75012).**



**Fig. 2.11 – Comparing measured and model Mass distributions (Unrestricted fit yielding Shape of 0.75012).**

Fitting of measured data to the Gamma distribution strives to conserve material. It is possible to conserve either mass or moles but not generally both. The procedure programmed into the Streamz software tries to minimize the

difference in the user specified material. The goodness of the fit can be visualized in a comparative molar distribution plot as shown in Fig. 2.9. This plot is for a specified Shape of 2.

The error bars are 5%. As is clear, the errors are quite big. The same comparison, but for an unrestricted regression, yields Fig. 2.10. The material being conserved in both cases is mass, while the material being compared is moles. If we compare mass fractions, as in Fig. 2.11, the results are even better. The minor differences are in the average molecular weight being calculated by the model.

#### 2.2.4 From the Gamma Model to Discrete Fractions

A conversion of petroleum stream from one characterization to another using the Gamma distribution model requires the breaking up of the calculated continuous model into discrete parts. The destination stream will be characterized by a number of components, each with a defined molecular weight (apart from other EOS properties). Since at this point the parameters of the Gamma model are already fixed (from the fit), the essence of the “split” is the determination of the boundary molecular weights separating each fraction and the amount of each fraction. One of the Gamma parameters, **bound**, already defines the lower bound MW of the first fraction. The upper boundary of the last fraction is always set at infinity, causing the area under the function (and thereby the sum of the normalized mole fractions) to equal 1. The boundaries of the other fraction are set according to the following:

- If user specified lower molecular weights for each fraction of the output characterization is available, they are used directly. Mole (or mass) fractions are calculated based on the amount of moles (or mass) determined for the continuous function within those bounds and the calculated molecular weights.
- In the absence of user specified lower molecular weights, arithmetic averages of fraction molecular weights are calculated and used as boundary molecular weights. The mole (or mass) fractions are then similarly obtained.

This procedure allows for conversion of multiple, but related, streams using individual Gamma distribution fit to the source stream into destination streams having the same Gamma distribution parameters. The destination stream will have defined average (or lower bound) molecular weights and the calculated mole fractions (or any other molar or mass quantity the program is asked to convert) are distributed in the manner consistent with it and the continuous distribution. Subsequent streams will typically vary in time, which will be reflected in the parameters calculated. But all destination streams will

correspond to the same defined average (or if chosen, lower bound) molecular weights. The molar or mass quantity calculated will reflect the fluid type.

### 2.2.5 TBP Stream Data Fit

One application of the Gamma distribution to petroleum streams is to fit true boiling point (TBP) data to the model. The source stream is the TBP data with its distillation “cuts” along with the mole or weight fractions. The destination stream could be a single carbon number (SCN) characterization with molecular weights from the Katz-Firoozabadi<sup>6</sup> table.

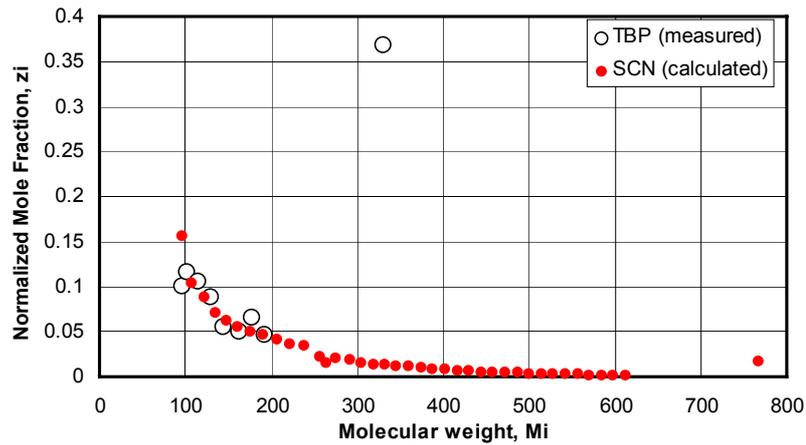


Fig. 2.12 – Application of Gamma distribution to TBP data fitting and conversion to SCN.

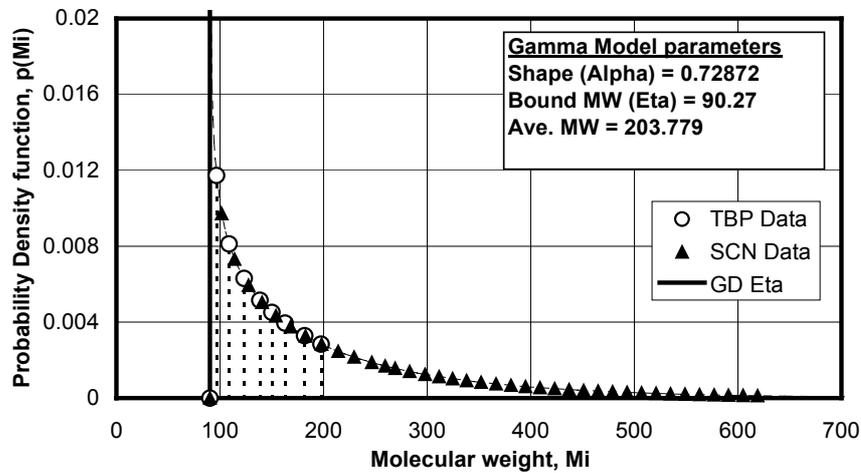


Fig. 2.13 – Molar distribution comparison of original (TBP) and converted (SCN) streams.

The main purpose here is the determination of Gamma distribution parameters, which can later be used to split C<sub>7+</sub> fractions into user specified number of fractions. Fig. 2.12 shows TBP data fit to the Gamma distribution, the plot of the calculated model, and the discrete lower MW points calculated for the output SCN stream. Fig. 2.1 shows the original TBP (measured) and SCN (calculated) normalized mole fractions vs. MW.

### 2.2.6 Splitting of a Single C<sub>7+</sub> Fraction

Many times petroleum fluid samples contain only the detailed descriptions of pure components (upto and including hexanes, which is frequently considered as a single component in EOS models). The information of the lumped C<sub>7+</sub> fraction is limited to the average molecular weight ( $M_{7+}$ ) and specific gravity ( $\gamma_{7+}$ ). For accurate prediction of phase behavior, EOS calculations required this single fraction to be split into 3 or more fractions. Without any other information, some PVT programs use Gaussian Quadrature to obtain the split. This is detailed by Whitson et.al.<sup>5</sup> in “C<sub>7+</sub> characterization”, where they also uses Gamma distribution to obtain the fit, and uses Gaussian Quadrature to obtain the molecular weight bounds and mole fractions.

If Gamma distribution parameters have previously been obtained from a TBP analysis of a related sample and the number of plus fractions and their average MW have been fixed, PSM software can be used directly to convert the single fraction stream to the desired characterization. In such a case, there is no “fit” of the input stream as all the regression parameters are fixed (previously obtained from a fit of the related TBP sample). The pre-determined model is used to split into the desired discrete characterization with user defined average molecular weights. Fig. 2.14 shows the results of PSM application for the split of a single C<sub>7+</sub> stream with average MW=203.78 into a stream with 9 C<sub>7+</sub> components also with an average MW=203.78.

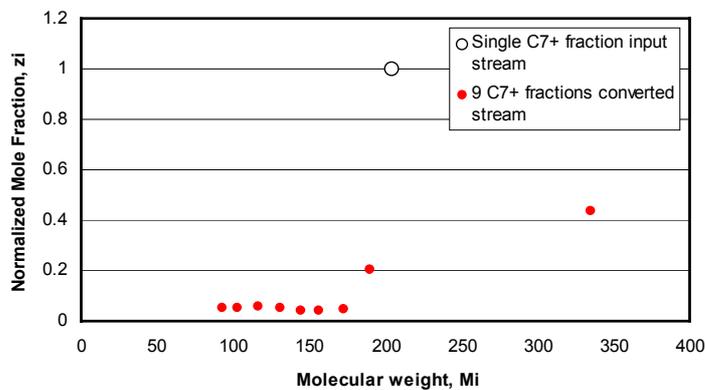


Fig. 2.14 – Split of single C<sub>7+</sub> fraction into 9 fractions.

### 2.2.7 Pseudoization of EOS Characterization

Development of an equation of state (EOS) model to represent fluid systems found in a petroleum reservoir is a frequent requirement. This involves multi-parameter regression to “tune” the initial default model to measured data. All available data (i.e. samples) should be used to tune the EOS. To enable the EOS to fit all data, usually more components are required (typically “pure” hydrocarbon components  $C_1$  to  $C_6$ , non-hydrocarbons like  $N_2$ ,  $CO_2$ ,  $H_2S$ , and minimum 5 heavy fractions). The actual EOS calculations in reservoir however cannot afford this “detailed” description and a pseudoization scheme is adopted to reduce the number of components to 5-8.

All current pseudoization schemes involve step-wise reduction of components with the matching of identified “data” at each step. Fevang et.al.<sup>1</sup> discuss and refer many schemes while developing their pseudoized EOS using this multi-step procedure. Many EOS parameters are modified at each step to match this set of data. Pseudoization usually means a direct “lumping” of all the components forming the lumped component for all the feeds. Sometimes this scheme results in the “wrong” phase being identified by the EOS calculation when the pseudoized EOS is used, from that when the full EOS is used. This might be expected in very near-critical fluids.

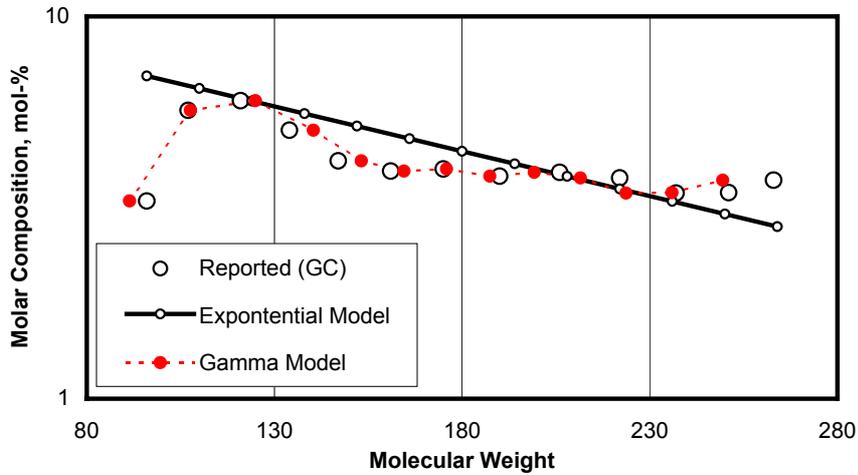
Lumping of fractions implies simple partitioning of pseudoized components with no over-lapping. Coats<sup>2</sup> acknowledges that grouping could allow for over-lapping of amount. Using Gamma fitting it is possible to allow over-lapping which would be a more realistic assumption. Practically this scheme is difficult to implement, as the Gamma distribution would have to be built into the PVT program. Recently Zick<sup>9</sup> has developed a new PVT program, **PhazeComp**<sup>10</sup>, that allows lumping of components using the Gamma Distribution while performing pseudoization.

### 2.2.8 Comparison of Exponential vs. Gamma Fit

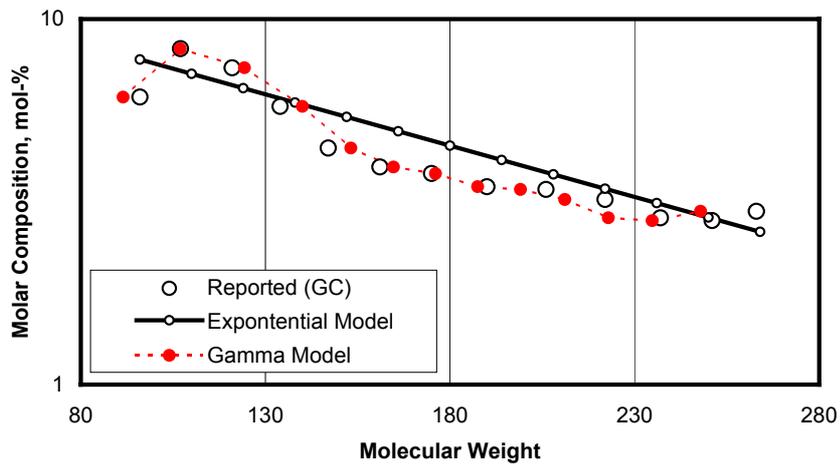
A field in UAE came up with 10 different samples, to be used for development of an EOS model. The  $C_{7+}$  fraction for all samples had GC analysis. For 9 of the 10 samples,  $C_{7+}$  were split into 14 fractions while the last into 6.

For purpose of splitting the  $C_{7+}$  fraction into any required number of fractions it is usually fit to a continuous model. Fitting to a simple exponential model is possible using the procedure outlined in the SPE Monograph<sup>6</sup>. This can be implemented in a spreadsheet. The same data is also fit using the Gamma distribution model. The superiority of the Gamma distribution over the exponential model is evident in Fig. 2.15 and Fig. 2.16, which are chosen from among the 10 samples. Fig. 2.15 depicts the results for sample 1 where the

differences between the two models were the maximum. Fig. 2.16 depicts the results for sample 5.



**Fig. 2.15 – Application of Gamma fit to related samples (1) showing a much better fit compared to the exponential model.**



**Fig. 2.16 – Application of Gamma fit to related samples (2) showing a much better fit compared to the exponential model.**

### 2.2.9 Typical Gamma Fit usage for Multiple Samples

As outlined previously, one method of splitting a C7+ fraction into a desired number of fractions (e.g. for development of an EOS for reservoir simulation) is to previously fit detailed analysis data (e.g. GC analysis) to a continuous model. Superiority of the Gamma distribution model has also been

demonstrated. A typical situation exists where multiple samples exist with very different characterizations (i.e. number of components and their molecular weights). They are to be fit to the Gamma distribution model with varying parameters but with the same  $\eta$  (i.e. lower molecular weight). A recommended procedure for PVTx is:

- Fit all samples restraining the **Origin** parameter to 1 (PVTx cannot handle the 4-parameter Gamma model). The results for a typical (North Sea) case are reported in Table 2.1.
- Take an average of the  $\eta$  values (=91.099). Use the minimum between this average and lowest 1<sup>st</sup> fraction molecular weight. The 5<sup>th</sup> sample has the minimum of 90.
- Re-fit all samples while restraining Origin (=1) and Bound (=90). The final parameters for the 5 samples are reported in Table 2.2.
- Split the original C<sub>7+</sub> fraction into desired characterization having the same number of fractions and associated MWs.

**Table 2.1 – Initial Gamma Distribution Fit to multiple samples, variable  $\eta$ .**

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Shape Parameter (Alpha)	0.94698	0.75041	0.70056	0.91588	0.70929
Average MW Multiplier	0.99997	1.00031	1.00031	0.99789	0.99896
Bounding MW Multiplier	0.96277	0.96468	0.99277	1	1
Origin MW Multiplier	1	1	1	1	1
Number of Iterations	3	4	4	4	3
Beta Parameter	27.984	180.025	182.215	145.5	190.558
Average Plus MW	118.926	223.843	220.973	224.26	225.162
Bounding MW	92.426	88.751	93.32	91	90
Origin MW	92.426	88.751	93.32	91	90
RMS % Error	0.221	0.706	0.588	0.879	0.587

**Table 2.2 – Final Gamma Distribution Fit to multiple samples, fixed  $\eta = 90$ .**

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Shape Parameter (Alpha)	1.09328	0.78101	0.74270	0.93613	0.70932
Average MW Multiplier	0.99982	0.99470	1.00016	0.99782	0.99896
Bounding MW Multiplier	0.93750	0.97826	0.95745	0.98901	1.00000
Origin MW Multiplier	1.00000	1.00000	1.00000	1.00000	1.00000
Number of Iterations	3	3	4	3	4
Beta Parameter	26.442	169.764	176.302	143.403	190.551
Average Plus MW	118.908	222.587	220.939	224.243	225.162
Bounding MW	90.000	90.000	90.000	90.000	90.000
Origin MW	90.000	90.000	90.000	90.000	90.000
RMS % Error	0.269	3.045	0.628	0.902	0.587

The results of the application of this procedure are depicted in Fig. 2.17 and Fig. 2.18. Fig. 2.17 is for sample 1 where the reported GC data were mole fractions. Fig. 2.18 contains similar plot for samples 2-5 where mass fractions were reported.

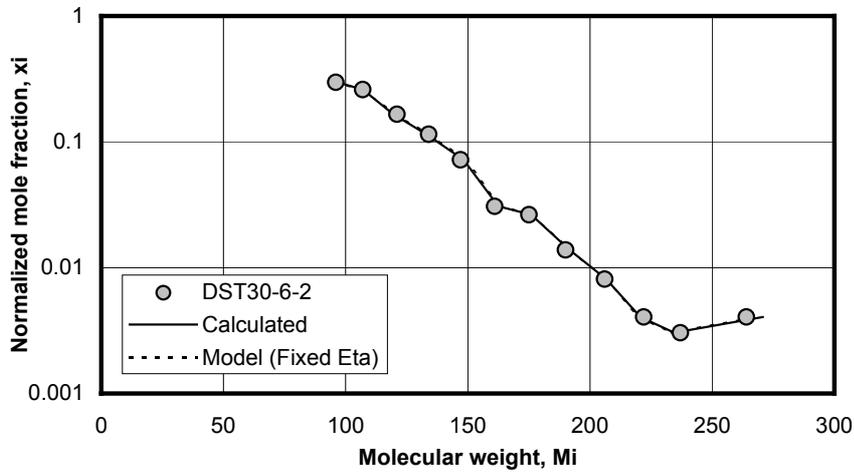


Fig. 2.17 – Application of Gamma fit to multiple samples (1).

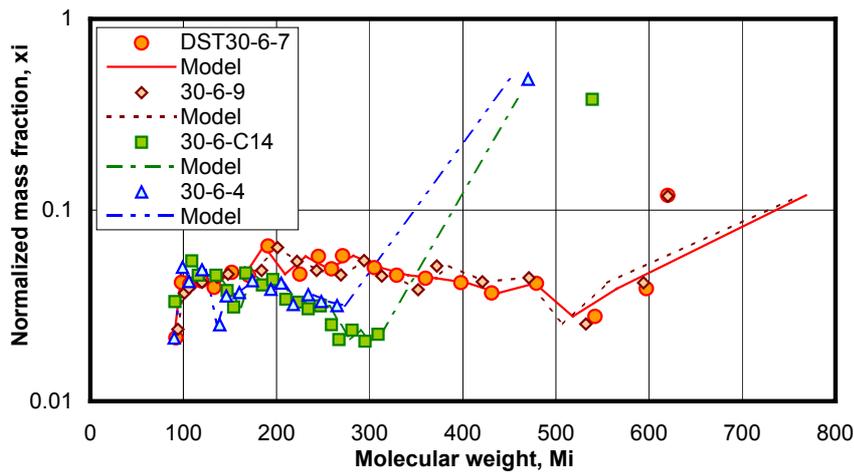


Fig. 2.18 – Application of Gamma fit to multiple samples (2).

### 2.2.10 Gamma Distribution Application to Reservoir Streams

Compositional Reservoir simulation is usually performed using a minimum number of components. Frequently the results are to be converted to a “full” EOS for process calculations. Use of Gamma distribution is a viable alternative to convert the  $C_{n+}$  fractions in the reservoir stream to equivalent

fractions in the process stream. This can be done on a time-step basis for each grid-cell connection, for maximum accuracy.

What are the limitations of such an application? What are the various options and methods of discretization? Are some methods more accurate than others? To answer these questions a relatively tough case was chosen and a suite of methods were applied to convert the pseudoized plus fractions into “full” plus fractions. While different methods could also be used for the “minus” fractions, that issue is not discussed in this section as the Gamma distribution only applies to the plus fractions.

A sector model was constructed and initialized with the fluid presented in Table 2.3. The pseudoized model had 9 components with only 3 plus fractions. The “full” model had 16 components with 6 plus fractions. The initial fluid in the 9c model was a simple lumping of the compositions of the 16c fluid. The fluids in place and the recovery in the two models are very close suggesting that the difference in the fluid system does not affect the simulator performance to any significant degree. In other words it is assumed that the two simulations produce “identical” fluids.

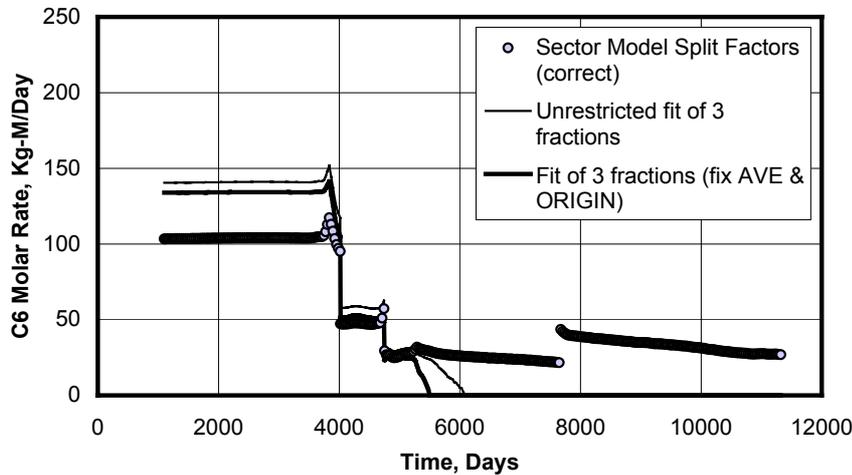
**Table 2.3 – Initial fluids in the 16-component and 9-component sector models and the details of component lumping.**

16-Component			9-Component		
Components (i)	Mw	Mole Fraction	Components (j)	Mw	Mole Fraction
N2	28.01	3.1553E-03	C1	16.09	4.4024E-01
CO2	44.01	6.8873E-03	CO2	44.01	6.8873E-03
C1	16.04	4.3708E-01	C2	30.07	6.1538E-02
C2	30.07	6.1538E-02	C3	44.10	4.7249E-02
C3	44.10	4.7249E-02	C4	58.12	3.3476E-02
IC4	58.12	9.4348E-03	C5	72.15	2.4545E-02
C4	58.12	2.4041E-02	C6F3	103.72	1.3008E-01
IC5	72.15	1.0215E-02	F4	165.41	1.0963E-01
C5	72.15	1.4329E-02	F5F6	364.75	1.4635E-01
C6	86.18	2.1530E-02	<b>Lumping of Components</b>		
C7+1	94.39	3.8533E-02	If j = 1 then i = 1, 3	If j = 2 then i = 2	
C7+2	106.24	3.4125E-02	If j = 3 then i = 4	If j = 4 then i = 5	
C7+3	121.20	3.5895E-02	If j = 5 then i = 6, 7	If j = 6 then i = 8, 9	
C7+4	165.41	1.0963E-01	If j = 7 then i = 10, 11, 12, 13		
C7+5	293.07	1.0927E-01	If j = 8 then i = 14	If j = 9 then i = 15, 16	
C7+6	588.62	3.7079E-02			

$$S_{ij} = z_i / z_j \dots\dots\dots (2.19)$$

Split factors were generated from the results of the 16c simulation by assuming the trivial lumping of the components as shown in shown in Table 2.3 using

Eq. (2.19). These split factors, as a function of  $C_{6+}$  content, were applied on the results of the 9c simulation. The result, plotted as circles in Fig. 2.19, Fig. 2.20, Fig. 2.21, and Fig. 2.23, is assumed to be the “correct” converted stream to which other conversions are compared. This assumption is, however, not entirely correct as discussed later in this section. Results of the first “plus” fraction  $C_6$  (where the differences were maximum) will only be discussed.

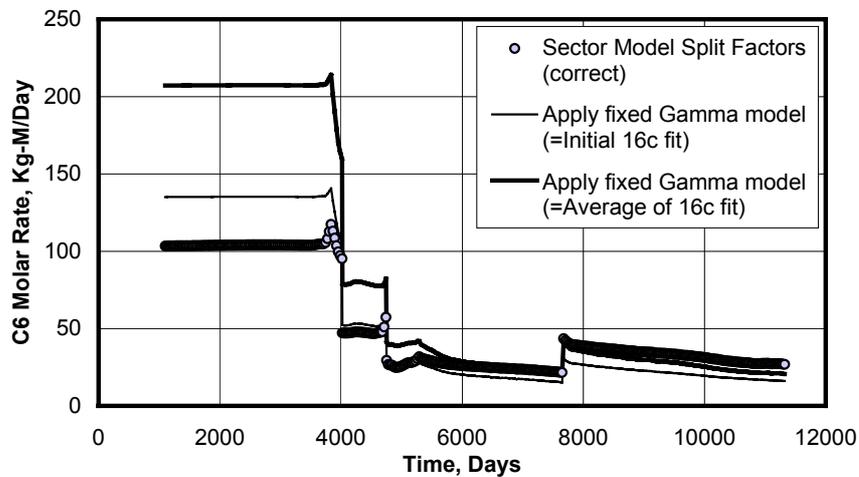


**Fig. 2.19 –Fit to Gamma model using only 3 fractions first using all four regression parameters and second using only two. Compared to the correct solution by use of Sector model split factors.**

The thin solid line in Fig. 2.19 is the result of applying the Gamma distribution to the 9c streams. The 9c streams contain only 3 plus fractions and the Gamma distribution itself has four parameters. Trying to find an optimal solution in this situation is prone to errors. However Streamz calculates reasonable results somewhat over-predicting the amounts of the first fraction ( $C_6$ ). After about 5000 days the program severely under-predicts the amounts of  $C_6$  going down to zero from 6000 onwards, where the actual amounts is in the range of 25-50 Kg-Mol/Day. This is because the fit of the 3 plus fractions to the Gamma distribution, containing too few information for an unique solution, forces the lower bound of the distribution to be equal to the average molecular weight of the first fraction of the 9c characterization ( $C_6$ ). This results in the zero amounts.

In general it is inadvisable to fit a function with number of parameters exceeding the number of data points. To avoid this problem, the two parameters **Average** and **Origin** were then fixed at a value of 1. These are reasonable constraints, as the average molecular weight of the input stream becomes invariant, and the value of **Origin** equal to 1 is merely the use of the classical Gamma distribution in preference to the default “truncated” one used

by Streamz. Imposing these constraints results in improved results plotted as thick lines in Fig. 2.19. While the over-prediction is reduced, the start of under-prediction and the “zero-amount” are hastened.



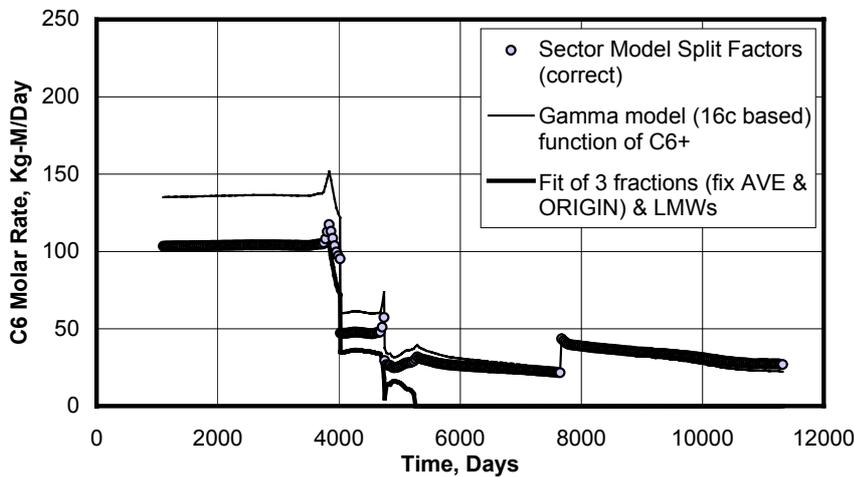
**Fig. 2.20 – Fit to fixed Gamma model first using the initial 16-component composition for the fit and second an average of fitting all 16-component streams. Compared to the correct solution by use of Sector model split factors.**

We see clearly that having too few information in the original stream (e.g. just 3 fractions as above), results in a very poor fit to the Gamma distribution model. We proceed further in our quest to obtain a better method for the application of Gamma distribution to reservoir streams. The initial fluid in the pseudoized and the full models are identical. Since we usually have a composition of the initial full fluid, we can use it to obtain a better Gamma model for the fluid.

The fit to the initial 16c stream was performed and resulted in Gamma model parameters: **Shape** 0.75074, **Average** 1.00000, **Bound** 84.009, and **Origin** 0.97621. By fixing these parameters when specifying the conversion of 9c streams, we basically force a fixed Gamma model to each stream and split it according to this single model and the requested output molecular weights. Even though the stream would best fit different Gamma models for each time-step, we force a single, but a better model. The resulting plot is the thin, solid lines in Fig. 2.20. It follows the thick solid lines of Fig. 2.19 upto about 5000 days, indicating that the Gamma fit of the 3 fraction (with **Ave** and **Origin** fixed) was probably as good as the fit of the 7 fractions. The major improvement is after this time when there is markedly less under-prediction. In fact the molar rate never goes to zero as in the previous cases.

In typical applications, there would typically be a single available composition in the “full” characterization for fitting to the Gamma model and its subsequent use to convert pseudoized streams. However, as in the present test case, we might have a comparable run of the detailed model. This is sometimes done on a portion of a full-field model (e.g. a sector model), for various sensitivities. The goodness of such a scaled-down model is gauged by its ability to reproduce the important features and characteristics of the full-field model. Based on this philosophy, Gamma distribution parameters obtained from a scaled-down model may be used with confidence with full-field models.

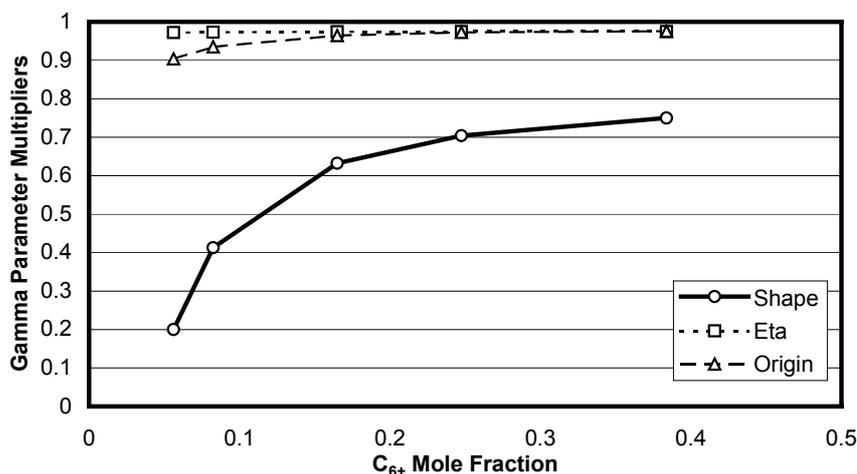
The approach is then to capture the variation of the Gamma parameters during the full simulation and use it, rather than the initial values as in the previous case. We have explored two possibilities: 1) Use an average of the parameters of the 16c fit, and 2) Use variable parameters as a function of some variable characterizing the variation of the streams. The result of using the first possibility is the thick solid line in Fig. 2.20. The over-prediction during the initial duration is extreme (about 100% to start with), but the agreement after about 6000 days is excellent.



**Fig. 2.21 – Fit to 16-component based Gamma model that is a function of  $C_{6+}$  amount. Compared to the correct solution by use of Sector model split factors and to a fit of 3 fractions while using lower molecular weights.**

One of the features of Streamz is its ability to allow all conversion parameters (split factors and/or gamma parameters) to be piece-wise linear functions of predefined variables. Whereas normal variables need to be predefined and must exist along with the stream quantities, compositional properties can be created on-the-fly and used for conversions. Using this feature, a *lumped*

mole fraction<sup>a</sup> amount was used as the control variable. When the 16c streams were fit to the Gamma distribution, the values of Gamma parameters  $\alpha$ ,  $\eta$  and  $\eta^o$  at initial, final, and three intermediate points (25, 50, and 75% of the extreme  $C_{6+}$  values) were specified. When this step-wise varying Gamma model, shown in Fig. 2.22, was applied on the 9c streams for splitting, the result is the thin solid line in Fig. 2.21. The agreement with the “correct” results is almost as good as any other during all time, while being the best during the end duration of simulation.

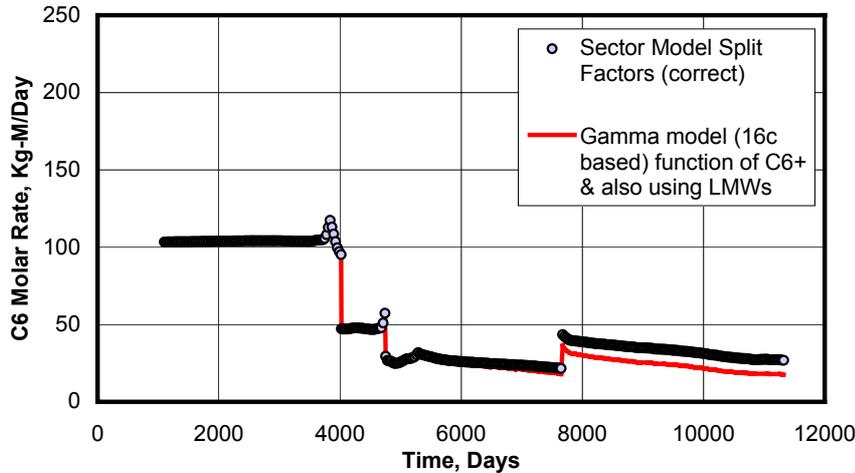


**Fig. 2.22 – Step-wise varying Gamma parameter multipliers, as a function of  $C_{6+}$  mole fraction, used in the split of 9c streams. The parameters were obtained by fitting the 16c streams to the Gamma model keeping Ave fixed at 1. Eta ( $\eta$ ) and Origin ( $\eta^o$ ) can be specified as either multipliers or actual MW values. For a multiplier of 1 in this case  $\eta = 86.178$  and  $\eta^o = \eta$ . Shape ( $\alpha$ ) is just a parameter.**

The results in Fig. 2.21 still over-predict about 30% during the initial period and slightly under-predict during the later period. Clearly the initial period is more critical as far as process design capacities are concerned. The reason for the differences is the inaccurate selection of molecular weight boundaries. While EOS characterizations are typically characterized by averaged MWs, the integration for calculating the amounts during discretization of the continuous Gamma distribution is done between molecular weight **boundaries** (defined using lower molecular weights LMW). The LMWs during discretization are determined as arithmetic averages of the supplied average MWs. This is the default way of fixing the LMWs and could be quite different from the inherent bounds in a split using split factors (based on trivial lumping). The locations of the LMWs determine the amount calculated for each fraction.

<sup>a</sup> As implemented in Streamz, the mole fraction of a lumped fraction is obtained by specifying its **Moles/Mole** property.

bounds in a split using split factors (based on trivial lumping). The locations of the LMWs determine the amount calculated for each fraction.

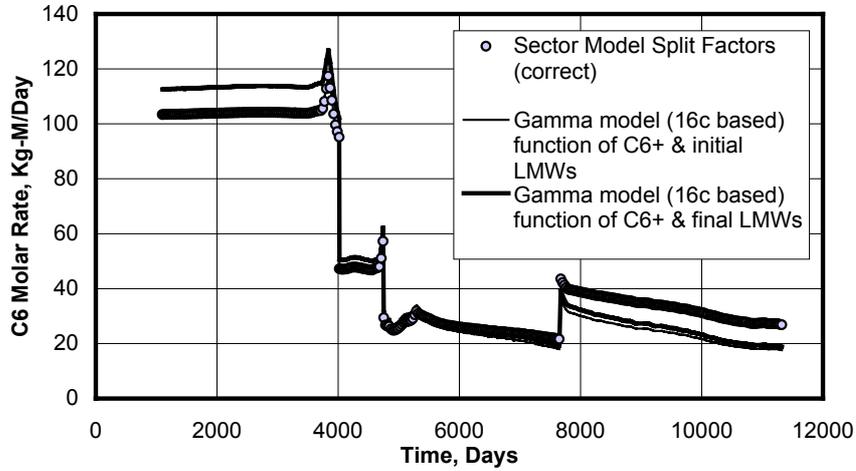


**Fig. 2.23 – Fit to 16-component based Gamma model that is a function of  $C_{6+}$  amount and at the same time using lower molecular weights for the split. Compared to the correct solution by use of Sector model split factors.**

Another method of specifying the bounds is to specify LMWs as part of the characterization. These have priority over the internally calculated averages of fraction MWs. A good estimate of these LMWs can be obtained from a fit of the original “full” fluid composition. The LMWs are also reported along with the Gamma parameters. When these LMWs are included in the characterization, the calculated molecular weight boundaries are more accurate and so are the calculated amounts. To test the effect of using the LMWs on a poor original match, it was used for the Gamma fit of just 3 plus fractions (with **Ave** and **Origin** being fixed). Although during the initial period the match is almost perfect, the under prediction and zero amount duration starts earlier and the degree is worse. This is clear from the thick solid line in Fig. 2.21.

The final plot (red solid line in Fig. 2.23) is the use of 16c-based Gamma Model as a function of  $C_{6+}$ . The dependence is implemented at a step-wise function specified at five equally spaced points. The intermediate values of Gamma parameters are interpolated (automatically by Streamz) based on the value of  $C_{6+}$  calculated for each stream (also automatically by Streamz). This results in the best overall match being near-perfect upto 6000 days. The slight, and progressively increasing, under prediction beyond this point is also within acceptable limits. The reason for this slight difference is the slightly different LMWs valid for the streams later on. Actually, the LMWs for each stream would be slightly different. Unfortunately for the single characterization

applicable for all the streams there is no way the LMWs could be specified to change. We see from the plot that the values of LMWs for the initial stream gives good results for the major, and the important, portion of the streams coming from the reservoir simulator.



**Fig. 2.24 – Effect of choice of LMW. Lower molecular weights used for the split are first set using the initial fluid composition and then using the final fluid composition (from the 16-component Sector model). Both compared to the correct solution.**

Fig. 2.24 is a plot showing the effect of the choice of LMWs. If LMWs from fit of the initial stream is chosen, the match is exact to about 5000 days after which the amounts are under-predicted. If the LMWs are taken from a fit of the final stream, the match is poor (over-prediction) during the initial period and only slightly less under-prediction during the later period.

Table 2.4 summarizes the various methods applied and discussed in this section. In general the choice of model parameters and the manner of specifying the Mw boundaries have great effect on the calculation of fraction amounts. It is always recommended to specify MW boundaries (LMWs) in the characterization to which streams are being converted. It is also recommended to use parameters from a fit of streams where more information is available (in this case 6 C<sub>n+</sub> fractions instead of 3 C<sub>n+</sub> fractions). However, if the streams being fit have sufficient information (4 or more C<sub>n+</sub> fractions) it is recommended to regress on all parameters “on-the-fly”.

**Table 2.4 – Methods of Applying Gamma Distribution for Splitting C<sub>7+</sub> of wellstreams.**

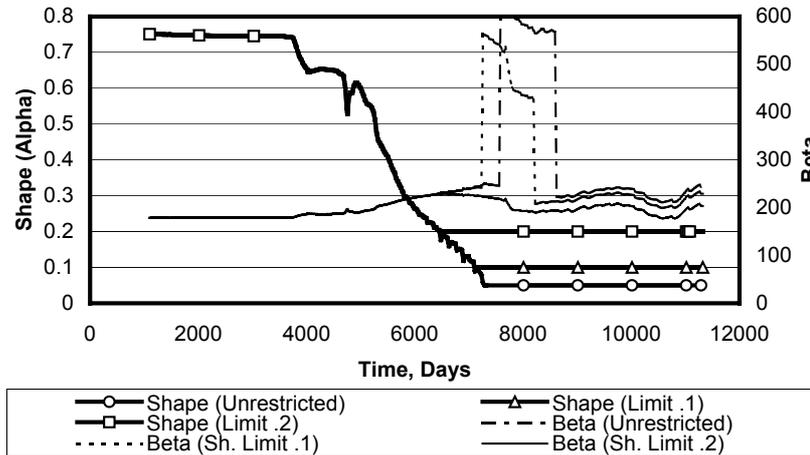
Method	Model Parameters	Mw Boundaries	Observation
Sector Model Split	-	-	Correct result for comparison

Factors			with other methods
Unrestricted Fit of 3 fractions (from 9c streams)	Regress ( $\alpha$ , $\eta$ , $\eta^0$ , $M_{n+}$ ) “on-the-fly”	Mean of fraction average MWs	Over-prediction by 40% initially; Zero amounts from 6000 days.
Fit of 3 fractions (fix <b>Ave</b> and <b>Origin</b> )	Regress ( $\alpha$ , $\eta$ ) “on-the-fly”; $\eta^0 = 1$ , $M_{n+} = 1$	Mean of fraction average MWs	Over-prediction by 30% initially; Zero amounts from 5500 days.
Apply fixed Gamma Model (= initial 16c fit)	Obtained from initial 16c fit; $\alpha = 0.75074$ , $\eta = 84.009$ , $\eta^0 = 0.97621$ , $M_{n+} = 1$	Mean of fraction average MWs	Over-prediction by 30% initially; Under-prediction by 50% later; No Zero amounts.
Apply fixed Gamma Model (= average of 16c fit)	Averages of parameters obtained from initial 16c fit; $\alpha = 0.41828269$ , $\eta = 84.7484883$ , $\eta^0 = 0.964624708$ , $M_{n+} = 1$	Mean of fraction average MWs	Over-prediction by 100% initially; Under-prediction by 5-10% later; No Zero amounts.
Gamma Model (16c based) a function of $C_{6+}$ amount	Step-wise varying parameters (obtained from initial 16c fit) as a function of $C_{6+}$ mole fraction	Mean of fraction average MWs	Over-prediction by 30% initially; Excellent agreement during later period.
Fit of 3 fractions (fix <b>Ave</b> and <b>Origin</b> )& LMWs	Regress ( $\alpha$ , $\eta$ ) “on-the-fly”	MW boundaries (LMWs) from fit of initial 16c feed	Excellent agreement initially; Zero amounts from 5500 days.
Gamma Model (16c based) a function of $C_{6+}$ amount & also using LMWs	Step-wise varying parameters (obtained from initial 16c fit) as a function of $C_{6+}$ mole fraction	MW boundaries (LMWs) from fit of initial 16c feed	Excellent agreement initially; Under-prediction by 10-30% later;
Gamma Model (16c based) a function of $C_{6+}$ amount & final LMWs	Step-wise varying parameters (obtained from initial 16c fit) as a function of $C_{6+}$ mole fraction	MW boundaries (LMWs) from fit of final 16c stream	Over-prediction by 10% initially; Under-prediction by 5-20% later;

Another reason for the poor performance during the later period is the use of fixed (non-varying) Gamma parameters below a certain value of  $C_{6+}$  due to observed behavior, described later, at low values of  $\alpha$ . The  $C_{6+}$  dependent Gamma parameters were defined for a range of  $C_{6+}$  starting from the maximum ( $=0.386065789$ ) to a minimum value where  $\alpha$  reaches its limit of 0.2 ( $C_{6+} = 00.056149675$ ). In the 16c streams this corresponds to 6479 days. All 9c streams having  $C_{6+}$  content less than this value (typically the later period), will use a fixed gamma model applicable at this limiting value. Thus the streams beyond the limit actually use an “incorrect” model to split the amounts, resulting in the poor match during this later period.

It was observed for this test case that an unconstrained fit of Gamma model to the 16c streams causes numerical problems for very low values of  $\alpha$ . Default limits of  $\alpha$  in Streamz are 0.4 to 5.0. The streams in this test case push the

value of  $\alpha$  below 0.4 during the later period. The user can increase the default limits of  $\alpha$  to 0.05 – 20. When those limits were specified, it was observed that the Gamma parameters break away from the trend. This is shown in Fig. 2.25 where values of Shape = 0.05 (thick lines with open circles) and = 0.1 (thick line with open triangles) result in Beta ( $\beta$ ) shooting off to very high values (thin dotted line and thin dot-dashed line) before returning to normal values. A value of 0.2 for Shape resulted in the normal expected behavior (thin solid line).



**Fig. 2.25 – Variation of gamma parameters showing un-acceptable results from the model at very low values of “Shape”.**

This behavior of behavior of Gamma modeling is being studied and currently it is recommend to follow Streamz’ default limits on Gamma parameters.

### 2.2.11 Gamma Distribution Formulation as Split Factors

The Gamma distribution procedure built into the Streamz software calculates the split factors for converting incoming streams into output streams. This allows the filling up of the heavy fraction portion of the generic conversion matrix formulation described in Section 2.1. This is then applied to the source streams to calculate the destination streams before writing out to the stream file.

In fact the split factors calculated by Streamz during the Gamma fit and split can themselves be written out for later use.

### 2.2.12 Conclusions

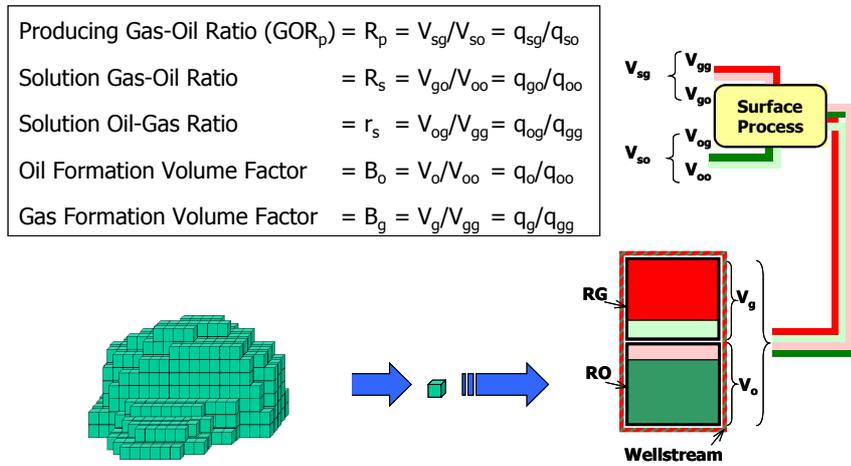
The use of the Gamma Distribution probability function for petroleum fluids has been explained. Its relationship to the molar distribution (mole fraction vs.

molecular weight) has been illustrated. The equations used to calculate the Gamma distribution has been explained along with the concept of the truncated Gamma Distribution used in Streamz.

Use of the Gamma distribution stream conversions has been illustrated with a host of typical applications. Improvement over the frequently used Exponential distribution has been shown. The limits of its usage to petroleum fluids and the effects of various parameters have been investigated and guidelines provided for proper use of the method.

### 2.3 BOz conversion

The BOz conversion method is based on original description of the compositional information contained in black-oil formulations given by Whitson and Brule<sup>6</sup>.



**Fig. 2.26 – Definition and explanation of terms used in the BOz conversion method.**

The modified black-oil formulation extends the classical black-oil concepts by including a surface oil component contained in the reservoir gas phase. Thus, similar to the solution gas in the reservoir oil quantified by the term R<sub>s</sub>, we also have solution oil in the reservoir gas quantified by the r<sub>s</sub> (or R<sub>v</sub>, used in various literature) term. These, and other terms involved in the BOz method, are depicted in Fig. 2.26 and defined below.

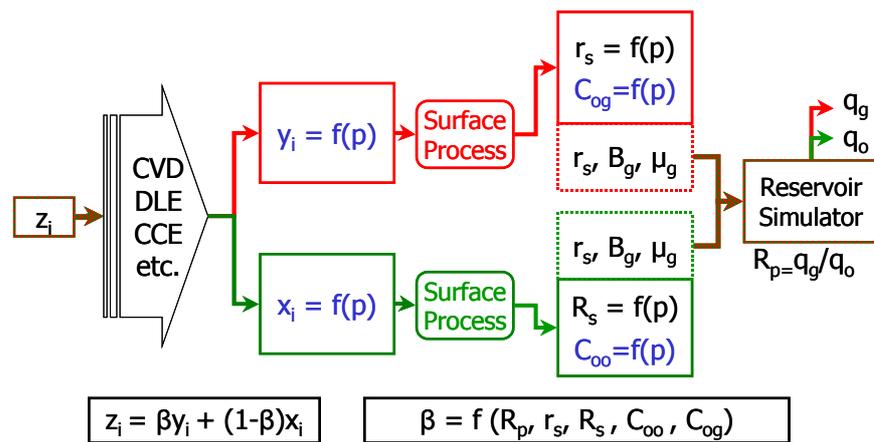
$$R_p = \frac{q_{sg}}{q_{so}} = \frac{q_{gg} + q_{go}}{q_{oo} + q_{og}} \dots\dots\dots(2.20)$$

$$R_s = \frac{q_{go}}{q_{oo}} \dots\dots\dots (2.21)$$

$$r_s = \frac{q_{og}}{q_{gg}} \dots\dots\dots (2.22)$$

BOz method converts results from a black-oil reservoir model characterization (with components surface oil,  $q_{so}$ , and surface gas,  $q_{sg}$ ) to a compositional EOS characterization. This conversion uses the fact that the PVT properties used by a BO simulator are generally prepared by an EOS-based PVT program.

The PVT program simulates depletion type experiments<sup>11</sup> at various pressures and takes both the equilibrium oil and equilibrium gas individually through a set of separators (surface process). Ratios of volumes of surface oil and surface gas define the BO parameters  $R_p$ ,  $R_s$ ,  $r_s$ ,  $B_o$ , and  $B_g$  as depicted in Fig. 2.26. Additional PVT properties  $C_{oo}$  and  $C_{og}$  and the equilibrium oil and gas compositions can also be reported but are generally not used. BOz method essentially recombines the equilibrium oil and gas compositions in proportions that are functions of PVT properties and the total wellstream GOR,  $R_p$ , which are themselves functions of pressure. This is pictorially depicted in Fig. 2.27.



**Fig. 2.27 – BOz conversion method is essentially the recombination of equilibrium oil and gas in proportions dependent on BO PVT properties.**

In its most fundamental form the BOz method is essentially a recombination of the reservoir oil and reservoir gas composition into a wellstream composition. This is similar to the recombination frequently performed on phase compositions from a well test to obtain the wellstream composition using the equation

$$z_i = \beta y_i + (1 - \beta)x_i \dots\dots\dots(2.23)$$

In well test the test separator GOR uniquely defines the vapor mole fraction  $\beta$ . But for the phases flowing in the reservoir it is not easy to estimate this fraction. The only measure of its value we have is the surface volume fractions  $R_p$ ,  $R_s$ , and  $r_s$ . Whitson and Brule<sup>6</sup> have detailed the development of this fraction, which they call  $F_g$ . It is essentially the mole fraction of the wellstream that comes from reservoir gas ( $n_g/(n_g+n_o)$ ).

$$\beta = \left[ 1 + \frac{F_{oo}(C_{oo} + R_s)}{(1 - F_{oo})(C_{og} + r_s^{-1})} \right]^{-1} \dots\dots\dots(2.24)$$

The term  $F_{oo}$  is the fraction of total stock tank oil coming from reservoir oil. It can be reduced to pressure related properties as shown:

$$F_{oo} = \frac{V_{oo}}{V_o} \dots\dots\dots(2.25)$$

Expanding  $V_{oo}$ ,

$$\begin{aligned} V_{oo} &= V_o - V_{og} \\ &= V_o - r_s V_{gg} \\ &= V_o - r_s (V_g - V_{go}) \dots\dots\dots(2.26) \\ &= V_o - r_s (V_g - R_s V_{oo}) \\ &= V_o - r_s V_g - r_s R_s V_{oo} \end{aligned}$$

Taking  $V_{oo}$  term to l.h.s.,

$$\begin{aligned} V_{oo} (1 - r_s R_s) &= V_o - r_s V_g \\ \frac{V_{oo}}{V_o} (1 - r_s R_s) &= 1 - r_s \frac{V_g}{V_o} \dots\dots\dots(2.27) \end{aligned}$$

$$\begin{aligned} F_{oo} &= \frac{V_{oo}}{V_o} = \frac{1 - r_s R_p}{(1 - r_s R_s)} \dots\dots\dots(2.28) \\ &= \frac{1}{(1 - r_s R_s)} - \frac{r_s R_p}{(1 - r_s R_s)} \end{aligned}$$

$C_{oo}$  and  $C_{og}$  are conversions from surface oil volume to an "equivalent" surface gas volume, defined as:

$$C_{ox} = K \frac{\gamma_{ox}}{M_{ox}} \dots\dots\dots (2.29)$$

where subscript "o" denotes surface oil, subscript "x" denotes either o (for reservoir oil) or g (for reservoir gas),  $\gamma$  is the specific gravity of surface oil from the relevant phase,  $M$  is the molecular weight of surface oil from the relevant phase, and  $k$  is a constant based on the unit system in use. This is detailed in Appendix B.

In general any two mixtures can be recombined using Eqs. (2.23), (2.24), and (2.28). If mixture A (with composition  $z_A$ ) has a surface GOR of  $R_A$  and mixture B (with composition  $z_B$ ) has a surface GOR of  $R_B$ , then the BOz method provides the total composition  $z$  of any mixture with  $R_A \leq \text{GOR} \leq R_B$ .

### 2.3.1 Split Factors for Black-oil to Compositional (BOz) Conversions

As described in Section 2.1, development of a general procedure for conversion from one characterization to another follows the generic matrix transformation:

$$u_i = \sum_{j=1}^m S_{ij} v_j, \quad i=1 \text{ to } n \dots\dots\dots (2.30)$$

In this equation, a characterization consisting of  $m$  components is converted into one consisting of  $n$  components. Each input component can theoretically partition into *none*, *some* or *all* of the output components. The split factors  $S_{ij}$  defines the fraction of the component  $j$  that goes into the particular output component  $i$ .

The development of the split factors for BOz conversion endeavors to come up with a matrix of split factors of the form:

$$m_i = \sum_{j=1}^2 S_{ij} q_j \dots\dots\dots (2.31)$$

We treat the surface oil and gas volumes as two components of a so-called black-oil characterization. Here the input characterization consists of only two

components,  $q_o$  and  $q_g$ . The output characterization consists of  $n$  components and corresponds to the EOS characterization used to generate the black-oil PVT table for the model whose results this BOZ conversion will transform.

In the equations that follow  $k$  is the constant volume (given by  $k = RT_{sc}/p_{sc}$ ) of one mole gas at standard conditions, which is independent of the type or source of gas. The value equals 23.69024531 ( $m^3/kmol$ ) in metric units and 379.4813268 ( $ft^3/lbmol$ ) in field units.

Using a basis of  $q_o$   $m^3$  of stock-tank oil, the equations for the moles of oil and gas can be derived as follows. We begin with the moles of surface oil from reservoir oil.

$$\begin{aligned} n_{oo} &= \frac{\text{Surface gas equivalent of } q_{oo}}{\text{Surface gas volume per mole}} \dots\dots\dots(2.32) \\ &= \frac{q_o F_{oo} C_{oo}}{k} \end{aligned}$$

For the moles of surface gas from reservoir oil we have,

$$\begin{aligned} n_{go} &= \frac{\text{Dissolved gas in } q_{oo}}{\text{Surface gas volume per mole}} \dots\dots\dots(2.33) \\ &= \frac{q_o F_{oo} R_s}{k} \end{aligned}$$

Moles of surface gas from reservoir gas is given by:

$$\begin{aligned} n_{gg} &= \frac{\text{Free surface gas volume } q_{gg}}{\text{Surface gas volume per mole}} \dots\dots\dots(2.34) \\ &= \frac{q_g - q_{go}}{k} \end{aligned}$$

Using the definitions for  $R_p$  (Eq. (2.20)) and  $R_s$  (Eq. (2.21))

$$\begin{aligned} n_{gg} &= \frac{q_o R_p - q_{oo} R_s}{k} \\ &= \frac{q_o R_p - q_o F_{oo} R_s}{k} \dots\dots\dots(2.35) \\ &= \frac{q_o (R_p - F_{oo} R_s)}{k} \end{aligned}$$

Finally, the moles of surface oil from reservoir gas is

$$\begin{aligned}
 n_{og} &= \frac{\text{Surface gas equivalent of } q_{og}}{\text{Surface gas volume per mole}} \\
 &= \frac{q_{og} C_{og}}{k} \dots\dots\dots (2.36) \\
 &= \frac{q_{gg} r_s C_{og}}{k} \\
 &= \frac{(q_g - q_{go}) r_s C_{og}}{k}
 \end{aligned}$$

Using the definitions for  $R_p$  and  $R_s$ ,

$$\begin{aligned}
 n_{og} &= \frac{(q_o R_p - q_{oo} R_s) r_s C_{og}}{k} \\
 &= \frac{(q_o R_p - q_o F_{oo} R_s) r_s C_{og}}{k} \dots\dots\dots (2.37) \\
 &= \frac{q_o (R_p - F_{oo} R_s) r_s C_{og}}{k}
 \end{aligned}$$

Writing down equations for moles of oil & gas, and substituting from Eqs. (2.32) to (2.37):

$$\begin{aligned}
 n_o &= n_{oo} + n_{go} \dots\dots\dots (2.38) \\
 n_g &= n_{og} + n_{gg}
 \end{aligned}$$

$$\begin{aligned}
 n_o &= \frac{q_o F_{oo} C_{oo}}{k} + \frac{q_o F_{oo} R_s}{k} \dots\dots\dots (2.39) \\
 &= \frac{q_o F_{oo} (C_{oo} + R_s)}{k}
 \end{aligned}$$

$$\begin{aligned}
 n_g &= \frac{q_o (R_p - F_{oo} R_s) r_s C_{og}}{k} + \frac{q_o (R_p - F_{oo} R_s)}{k} \dots\dots\dots (2.40) \\
 &= \frac{q_o (R_p - F_{oo} R_s) (r_s C_{og} + 1)}{k}
 \end{aligned}$$

Substituting Eq. (2.28) in Eq. (2.39), we obtain for moles of oil,  $n_o$ :

$$\begin{aligned}
 n_o &= \frac{q_o \left( \frac{1}{(1-r_s R_s)} - \frac{r_s R_p}{(1-r_s R_s)} \right) (C_{oo} + R_s)}{k} \\
 &= q_o \frac{(C_{oo} + R_s)}{k(1-r_s R_s)} - q_o R_p \frac{r_s (C_{oo} + R_s)}{k(1-r_s R_s)} \quad \dots\dots\dots(2.41) \\
 &= q_o \frac{(C_{oo} + R_s)}{k(1-r_s R_s)} - q_g \frac{r_s (C_{oo} + R_s)}{k(1-r_s R_s)}
 \end{aligned}$$

Substituting Eq. (2.28) into Eq. (2.40), we obtain for moles of gas,  $n_g$ :

$$\begin{aligned}
 n_g &= \frac{q_o (R_p - F_{oo} R_s) (r_s C_{og} + 1)}{k} \\
 &= \frac{q_o R_p (r_s C_{og} + 1) - q_o \left( \frac{1}{(1-r_s R_s)} - \frac{r_s R_p}{(1-r_s R_s)} \right) R_s (r_s C_{og} + 1)}{k} \quad \dots\dots\dots(2.42)
 \end{aligned}$$

$$\begin{aligned}
 n_g &= q_g \frac{(r_s C_{og} + 1)}{k} - q_o \frac{R_s (r_s C_{og} + 1)}{k(1-r_s R_s)} + q_g \frac{r_s R_s (r_s C_{og} + 1)}{k(1-r_s R_s)} \\
 &= q_g \frac{(r_s C_{og} + 1) [(1-r_s R_s) + r_s R_s]}{k(1-r_s R_s)} - q_o \frac{R_s (r_s C_{og} + 1)}{k(1-r_s R_s)} \quad \dots\dots\dots(2.43) \\
 &= q_g \frac{(r_s C_{og} + 1)}{k(1-r_s R_s)} - q_o \frac{R_s (r_s C_{og} + 1)}{k(1-r_s R_s)}
 \end{aligned}$$

Component moles are the product of the total phase moles and the phase mole fractions:

$$\begin{aligned}
 n_{oi} &= n_o x_i \\
 n_{gi} &= n_g y_i \quad \dots\dots\dots(2.44)
 \end{aligned}$$

The total moles of the wellstream are then the sum of the component moles of oil and gas:

$$n_i = n_{oi} + n_{gi} \quad \dots\dots\dots(2.45)$$

Substituting from Eqs. (2.41), (2.43), and (2.44) into Eq. (2.45),

$$n_i = \left[ q_o \frac{(C_{oo} + R_s)}{k(1 - r_s R_s)} - q_g \frac{r_s (C_{oo} + R_s)}{k(1 - r_s R_s)} \right] x_i + \left[ q_g \frac{(r_s C_{og} + 1)}{k(1 - r_s R_s)} - q_o \frac{R_s (r_s C_{og} + 1)}{k(1 - r_s R_s)} \right] y_i \quad \dots \quad (2.46)$$

Grouping together the  $q_o$  and  $q_g$  terms:

$$n_i = \left[ \frac{(C_{oo} + R_s)}{k(1 - r_s R_s)} x_i - \frac{R_s (r_s C_{og} + 1)}{k(1 - r_s R_s)} y_i \right] q_o + \left[ \frac{(r_s C_{og} + 1)}{k(1 - r_s R_s)} y_i - \frac{r_s (C_{oo} + R_s)}{k(1 - r_s R_s)} x_i \right] q_g \quad \dots \quad (2.47)$$

If we group the terms for  $q_g$  and  $q_o$  together, and choose to represent these pressure dependent factors as  $S_{ig}$  and  $S_{io}$  respectively, then  $n_i$  can be written as:

$$n_i = S_{io} q_o + S_{ig} q_g \quad \dots \quad (2.48)$$

This can be generalized as:

$$n_i = \sum_{j=1}^2 S_{ij} q_j, \text{ where } q_1 = q_o \text{ and } q_2 = q_g \quad \dots \quad (2.49)$$

with the following expansion for the split-factors:

$$S_{i1} = \frac{(C_{oo} + R_s)}{k(1 - r_s R_s)} x_i - \frac{R_s (r_s C_{og} + 1)}{k(1 - r_s R_s)} y_i \quad \dots \quad (2.50)$$

$$S_{i2} = \frac{(r_s C_{og} + 1)}{k(1 - r_s R_s)} y_i - \frac{r_s (C_{oo} + R_s)}{k(1 - r_s R_s)} x_i \quad \dots \quad (2.51)$$

All parameters making up the split-factors are pressure dependent. They are in fact known properties calculated by an EOS-based PVT program when it simulates a depletion type (CVD, DLE, CCE/CME) experiment. Simulation of a depletion type experiment is the standard method<sup>11</sup> for calculation of black-oil PVT tables for a black-oil simulator. Hence any EOS-based PVT program can generate the quantities in Eqs. (2.50) and (2.51) at the same time

it generates the black-oil PVT tables. A standard format for use with PSM/BOz has been defined for any PVT program. This format called “File format for extended black-oil PVT data” is included as Appendix B.

These split factors being functions of pressure dependent properties, themselves vary with pressure. To correctly capture this dependence, multiple tables of split factors should be specified as part of the full conversion table, each table for a particular node pressure. This node pressure corresponds to the depletion stage pressure defined in the experiment. A range of node pressures covering the range of pressure expected in the reservoir model should be used for accurate conversions.

### 2.3.2 Negative Split Factors?

The usage of split factors, developed in Section 2.3.1, for BOz stream conversion reveals the fact that they can be negative for some of the output components. Intuitively we feel that split factors cannot be negative, i.e. an input component cannot split into a negative amount of the output component. The key point here is that split factors themselves do not represent the amount of the output component but that the amount of the output component is a split-factor weighted sum of the input components:

$$m_i = \sum_{j=1}^2 S_{ij} q_j \dots\dots\dots(2.52)$$

For BOz conversions

$$m_i = S_{io} q_o + S_{ig} q_g \dots\dots\dots(2.53)$$

For each component there are two split factors, one for the oil rate and the other for the gas rate ( $S_{io}$  and  $S_{ig}$ ). We base our discussions on the assumption that if both  $q_g > 0$  and  $q_o > 0$ , then  $m_i > 0$  for all  $i$ . However  $S_{io}$  and  $S_{ig}$  maybe  $>$  or  $<$  0. A negative split factor for one (e.g.  $S_{io}$ ) is always compensated by the positive split factor for the other (e.g.  $S_{ig}$ ). In addition the product of the positive split factor and the corresponding amount should exceed the other product, to guarantee positive rates  $m_i$ .

Apart from this explanation of negative split-factors being consistent with positive physical molar rates, we now investigate the reasons for the negative split-factors. We start with the expression for the split-factors:

$$S_{ig} = \frac{(1 + r_s C_{og})}{k(1 - r_s R_s)} y_i - \frac{r_s (C_{oo} + R_s)}{k(1 - r_s R_s)} x_i \dots\dots\dots(2.54)$$

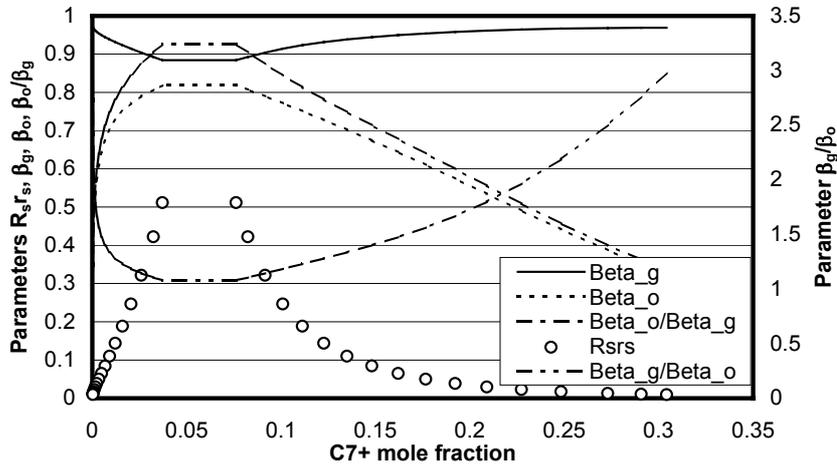
$$S_{io} = \frac{(C_{oo} + R_s)}{k(1 - r_s R_s)} x_i - \frac{R_s(1 + r_s C_{og})}{k(1 - r_s R_s)} y_i \dots\dots\dots (2.55)$$

Working with the oil split-factor,  $S_{io}$ , only and taking the common terms outside the bracket,

$$S_{io} = \frac{1}{k(1 - r_s R_s)} [(C_{oo} + R_s)x_i - R_s(1 + r_s C_{og})y_i] \dots\dots\dots (2.56)$$

$$S_{io} = \frac{R_s}{k(1 - r_s R_s)} \left[ \left( \frac{C_{oo}}{R_s} + 1 \right) x_i - (1 + r_s C_{og}) y_i \right] \dots\dots\dots (2.57)$$

The range of  $r_s R_s$  is from 0 to 1 (refer Fig. 2.28) so the term  $(1 - r_s R_s)$  is always positive. The constant  $k$  and the physical property  $R_s$  are also always positive. Hence the terms within the square brackets only must contribute to a negative split-factor.



**Fig. 2.28 – Variation of “negative criteria” parameters as a function of fluid type.**

Using the definition<sup>6</sup> of mole fraction of surface gas in reservoir phases:

$$\beta_o \equiv \frac{n_{go}}{n_o} \equiv \left( 1 + \frac{C_{oo}}{R_s} \right)^{-1} \dots\dots\dots (2.58)$$

$$\beta_g \equiv \frac{n_{gg}}{n_g} \equiv (1 + r_s C_{og})^{-1} \dots\dots\dots(2.59)$$

we obtain:

$$S_{io} = K \left[ \frac{x_i}{\beta_o} - \frac{y_i}{\beta_g} \right] \dots\dots\dots(2.60)$$

Here the constant  $K$  is the combination of factors/parameters that are necessarily positive, resulting in  $K$  itself being always positive. The split factor is negative if:

$$\left[ \frac{x_i}{\beta_o} - \frac{y_i}{\beta_g} \right] < 0 \dots\dots\dots(2.61)$$

$$\frac{x_i}{\beta_o} < \frac{y_i}{\beta_g} \dots\dots\dots(2.62)$$

$$\frac{\beta_g / \beta_o}{y_i / x_i} < 1 \dots\dots\dots(2.63)$$

$$\frac{\beta_g / \beta_o}{K_i} < 1 \dots\dots\dots(2.64)$$

$$K_i > \frac{\beta_g}{\beta_o} \dots\dots\dots(2.65)$$

While  $K_i$  is the familiar equilibrium  $K$ -value of the component  $i$ , the term  $\beta_g/\beta_o$  is a kind of “gross”  $K$ -value of the surface components<sup>6</sup>  $y_g$  (gas from gas) and  $x_g$  (gas from oil). Hence the oil split factors will be negative if the component  $k$ -value in the reservoir phases is greater than the “gross”  $k$ -value of the surface components in the reservoir gas phase. Variation of parameters  $\beta_g$ ,  $\beta_o$ ,  $\beta_g/\beta_o$  and  $\beta_o/\beta_g$  as a function of fluid type is depicted in Fig. 2.28. It is clear that these parameters are smoothly varying functions of the fluid type ( $C_{7+}$  mole fraction used here to characterize the fluid type).

Similar development results in the following expression for the criteria for negative *gas* split factors:

$$\frac{K_i}{(\beta_g / \beta_o) R_s r_s} < 1 \dots\dots\dots (2.66)$$

$$K_i < (\beta_g / \beta_o) R_s r_s \dots\dots\dots (2.67)$$

The terms  $R_s$  and  $r_s$  are essentially reciprocals of each other and their product is dimensionless ranging from 0 to 1. Hence the gas split factors will be negative if the component k-value in the reservoir phases is less than the scaled (reduced) “gross” K-value of the surface components in the reservoir gas phase. The reduction is due to the combined effect of  $R_s r_s$ , which is constant for a particular fluid at a certain pressure.

Another way of expressing the right hand side of Eq. (2.67) is to expand the terms thus:

$$\begin{aligned} \beta_g r_s &= \frac{r_s}{1 + C_{og} r_s} \dots\dots\dots (2.68) \\ &= \frac{1}{r_s^{-1} + C_{og}} \end{aligned}$$

$$\begin{aligned} \frac{R_s}{\beta_o} &= R_s (1 + C_{oo} / R_s) \dots\dots\dots (2.69) \\ &= R_s + C_{oo} \end{aligned}$$

Combining,

$$\beta_g r_s \frac{R_s}{\beta_o} = \frac{R_s + C_{oo}}{r_s^{-1} + C_{og}} \dots\dots\dots (2.70)$$

Hence the criteria becomes:

$$\frac{K_i}{\left( \frac{R_s + C_{oo}}{r_s^{-1} + C_{og}} \right)} < 1 \dots\dots\dots (2.71)$$

$$K_i < \left( \frac{R_s + C_{oo}}{r_s^{-1} + C_{og}} \right) \dots\dots\dots (2.72)$$

Looking only at the right hand side of Eq. (2.72),  $(R_s+C_{oo})$  is the total gas (surface gas and gas “equivalent” of surface oil) from reservoir oil. The term  $(r_s^{-1}+C_{og})$  is the total gas (surface gas and gas “equivalent” of surface oil) from reservoir gas. The ratio is, in a sense, the ratio of surface product volume from reservoir oil to the surface product volume from reservoir gas. In this sense it is also a “gross” partitioning of reservoir phases into surface components.

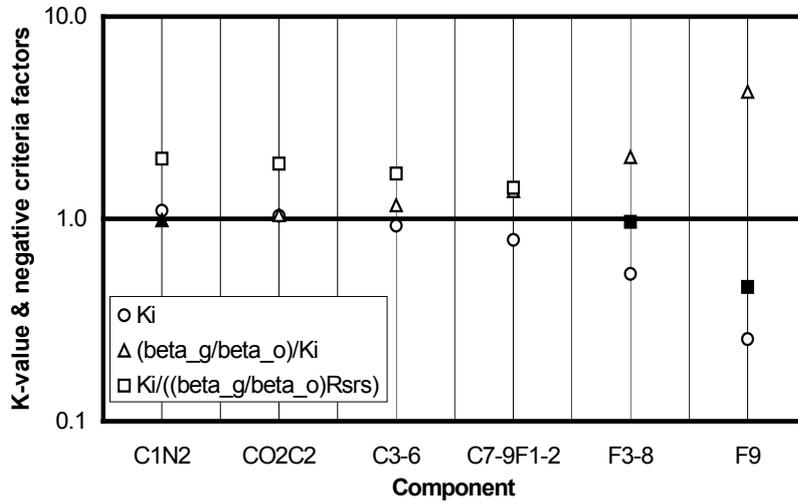


Fig. 2.29 – Variation of K-values and “negative criteria factors” for a 6-component system for an initial fluid.

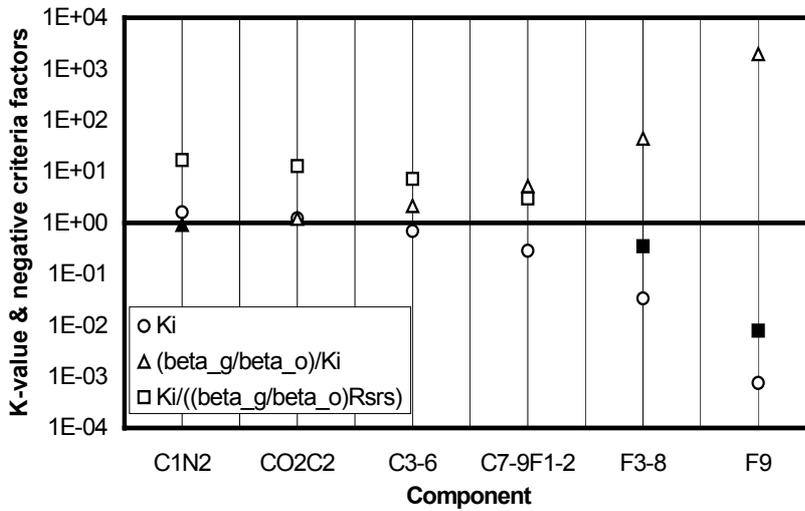
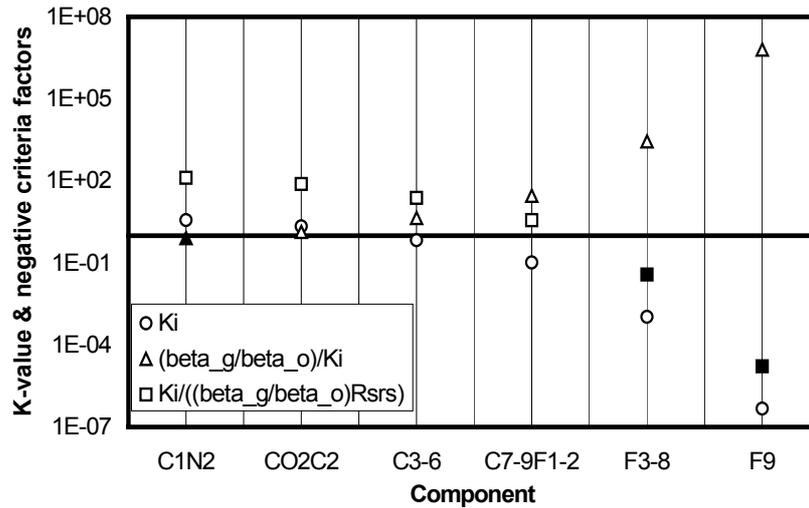


Fig. 2.30 – Variation of K-values and “negative criteria factors” for a 6-component system for a fluid depleted to mid-way stage.

Fig. 2.29 shows the two “negative criteria factors” given by Eqs. (2.64) and (2.66) plotted for each component. The plot also shows the trend of the component k-values. Fig. 2.30 and Fig. 2.31 are similar plots for depleted pressures at mid-way stage and end stage respectively. In these figures the circle symbols represent the component K-values. Triangles represent the “negative oil split factor criteria” parameter, and the squares represent the “negative gas split factor criteria” parameter. The filled triangles and squares denote the points at which these criteria parameters become less than 1 resulting in the split factors for those components to be negative.



**Fig. 2.31 – Variation of K-values and “negative criteria factors” for a 6-component system for a fluid depleted to end stage.**

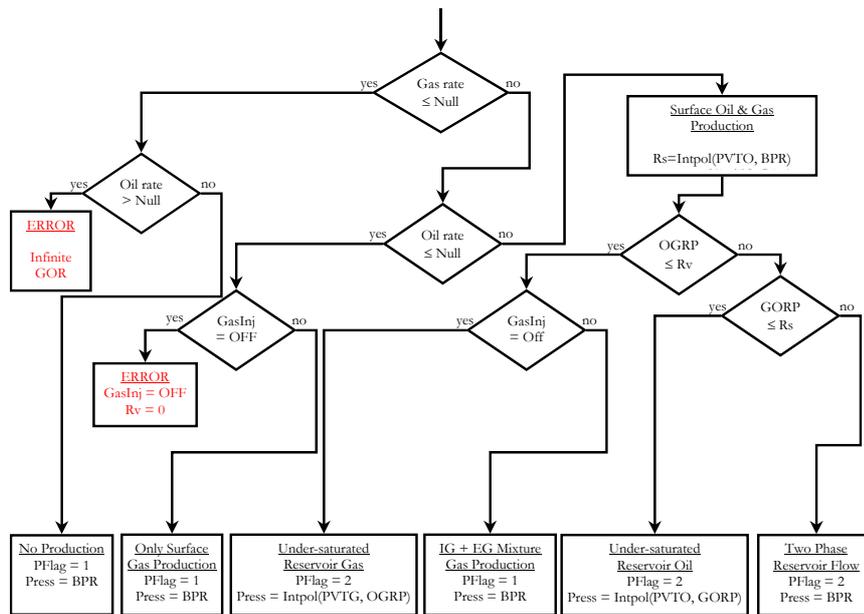
### 2.3.3 Implementation of Gas Injection

As explained in section 2.3, the BOz method provides the total composition of any mixture using its surface GOR. If the mixture is a combination of a “phase” A (with composition  $z_A$  and surface GOR  $R_A$ ) and another “phase” B (with composition  $z_B$  and surface GOR  $R_B$ ) then the BOz method is applicable for any combination of the two “phases” (i.e. within the range  $R_A \leq \text{GOR} \leq R_B$ ). Section 2.3.1 implements the BOz method into the split factor formulation used by Streamz allowing streams with surface oil and gas rates (i.e. essentially the  $R_p$ ) to be converted into component molar rates (i.e. essentially the mixture composition).

In the typical usage of the BOz method “phase” A is the reservoir oil (with composition  $x_i$  and GOR  $R_s$ ) and “phase” B is the reservoir gas (with composition  $y_i$  and GOR  $1/r_s$ ). The method is then applicable within the range of producing GOR  $R_s \leq R_p \leq r_s^{-1}$ .

When gas is injected into the reservoir, a third “phase” (say C) exists. If an injection well starts to produce or if gas breaks through, the producing GOR,  $R_p$ , is no longer within the range  $R_s$  to  $r_s^{-1}$ . If  $R_p$  becomes greater than  $1/r_s$  then the wellstream can be a mixture of a) B+C, b) A+C, or c) A+B+C. As implemented in PSM, a mixture of B (reservoir gas with composition  $y_i$  and surface GOR  $1/r_s$ ) and C (injection gas with a constant composition  $w_i$  and surface GOR = 0) is assumed to flow and the BOz method applied to this mixture separately.

The BOz method supplies two sets of tables for each mixture and the black-oil simulator post-processor decides which table is to be used based on the value of the current producing GOR,  $R_p$ . The actual logic used by the post-processor is depicted in Fig. 2.32.



**Fig. 2.32 – Logic for the implementation of Gas Injection case for BOz conversions.**

### 2.3.4 Error in Oil Rates

One of the checks (see Section 2.3.6) applied to the results of BOz conversion is to re-convert back to surface oil and gas rates by using an identical surface separation scheme used to generate the PVT tables in the first place. If a 3-stage separation system was used, the resulting molar streams are sent through the same process using Streamz **process** command.

To investigate the degree of error expected between the Black-oil oil rates and the PSM (generated + processed) oil rates, quantities are defined in Table 2.5.

This quantifies the error in processing gas and oil phases separately through surface separator (as done by the BO simulator indirectly through the use of BO PVT tables) versus processing the actual wellstream. PSM/Streamz re-processing is a method to correct for these errors, which can be up to 5-10%<sup>12</sup>.

We investigate the two mixtures implemented in the BOz method: 1) Mixture of EG and EO (normal case with OGR between  $R_v$  of EG and  $1/R_s$  of EO), and 2) Mixture of EG and IG (OGR ranging from  $R_v$  of EG and  $R_v$  of IG=0). The injection gas was a typical separator gas stripped off its  $C_{3+}$  content.

**Table 2.5 – Definition of term for evaluation of Error in Oil rates.**

EO	:	Equilibrium (reservoir) oil
EG	:	Equilibrium (reservoir) gas
IG	:	Injection gas
OGR <sub>BO</sub>	:	Producing oil-gas ratio for a BO stream (oil and gas rates) input into PSM (STB of oil per Mscf of gas)
OGR <sub>PSM</sub>	:	Inverse of gas-oil ratio from processing the wellstream resulting from PSM conversion of the BO stream
Error Parameter	:	OGR <sub>PSM</sub> /OGR <sub>BO</sub> . Equals 1 for zero error.

For each of the 2 mixtures a series of streams (surface oil, SO and surface gas, SG) were generated for the proper range of OGR by keeping a fixed SG and obtaining SO as  $SO = SG \times OGR$ . For example, for a mixture of IG and EG the streams are reproduced in Table 2.6.

**Table 2.6 – Computed input streams for checking of errors between original OGR and PSM converted and re-processed OGR. These values are for mixtures of EG and IG at a pressure of 3415.69 psia.**

OGR (STB/Mscf)	Volumes SO (STB)	SG (Mscf)
4.528800E-02	4.528800E+01	1000
4.075920E-02	4.075920E+01	1000
3.623040E-02	3.623040E+01	1000
3.170160E-02	3.170160E+01	1000
2.717280E-02	2.717280E+01	1000
2.264400E-02	2.264400E+01	1000
1.811520E-02	1.811520E+01	1000
1.358640E-02	1.358640E+01	1000
9.057600E-03	9.057600E+00	1000
4.528800E-03	4.528800E+00	1000
2.264400E-03	2.264400E+00	1000
2.264400E-04	2.264400E-01	1000
1.132200E-04	1.132200E-01	1000
4.528800E-05	4.528800E-02	1000
4.528800E-06	4.528800E-03	1000
4.528800E-08	4.528800E-05	1000
4.528800E-10	4.528800E-07	1000
0.000000E+00	0.000000E+00	1000

These streams were converted using Streamz to obtain a corresponding series of streams for the 15-component characterization defined. Five from these sets of compositions are reproduced in Table 2.7.

**Table 2.7 – Select five PSM output molar streams for indicated  $R_v$ . These values are for mixtures of EG and IG at a pressure of 3415.69 psia.**

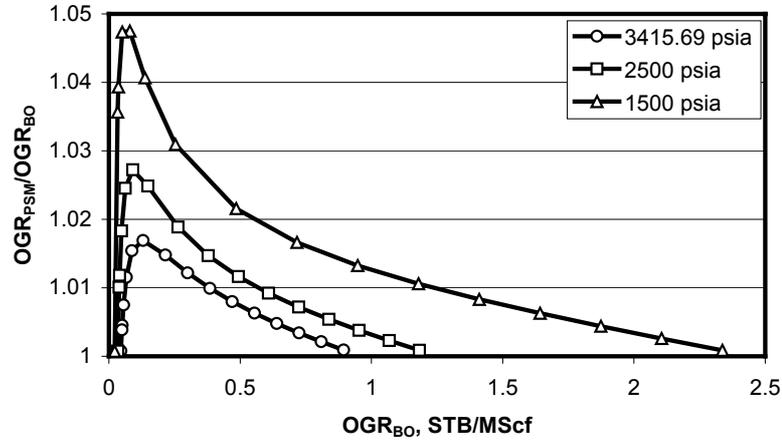
Component	$R_v =$ 0.045288	$R_v =$ 0.031702	$R_v =$ 0.018115	$R_v =$ 0.004529	$R_v =$ 0
N2	35.9565	31.2263	26.4961	21.7659	20.1892
CO2	232.681	237.625	242.57	247.515	249.163
C1	1791.69	1879.02	1966.36	2053.69	2082.8
C2	237.244	250.976	264.708	278.44	283.018
C3	189.808	132.866	75.9231	18.9808	0
IC4	40.6707	28.4695	16.2683	4.06707	0
C4	52.7992	36.9594	21.1197	5.27992	0
IC5	17.1861	12.0302	6.87442	1.71861	0
C5	22.0166	15.4116	8.80662	2.20166	0
C6	29.4161	20.5913	11.7664	2.94161	0
C7(1)	26.5305	18.5713	10.6122	2.65305	0
C7(2)	32.6794	22.8756	13.0718	3.26794	0
C7(3)	19.0198	13.3139	7.60794	1.90198	0
C7(4)	1.50272	1.05191	0.601089	0.150272	0
C7(5)	0.049945	0.034962	0.019978	0.004995	0

These series of compositions were automatically processed through an identical process using Streamz **process** command where the ratio of the resulting oil and gas rates gives the  $OGR_{PSM}$ . This, in essence, is the “process” OGR obtained by a sending a full mixture composition through a surface process, and is accordingly more correct than the  $OGR_{BO}$ .  $OGR_{BO}$  is the oil rate and gas rate obtained by sending separately the reservoir oil and gas through the surface separator (indirectly by the use of ECL PVT tables). The resulting oil-from-oil and oil-from-gas are added to get the total surface oil, SO and similarly for the total surface gas, SG.  $OGR_{BO}$  is thus SO/SG. Since separate processing is approximating the behavior of the process to the mixture, this is a less correct OGR.

As the PSM processing reproduces the EO and EG compositions at the extremes, a plot of the Error Parameter vs. OGR will have zero values at these points. Intervening mixtures of EO and EG result in an  $OGR_{PSM}$  that is higher than  $OGR_{BO}$ .

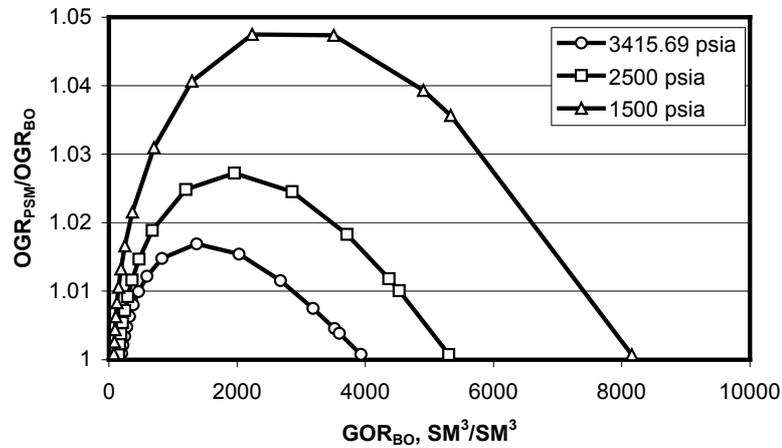
The experiment was conducted for reservoir phases at 3 different pressures of 3415.69, 2500 and 1500 psia. The injection gas composition at each pressure is identical while the reservoir gas and reservoir oil compositions vary with pressure. Fig. 2.33 shows the trend of the Error Parameter ( $OGR_{PSM}/OGR_{BO}$ ) vs.  $OGR_{BO}$  for mixtures of EO and EG. For pure EO and EG the Error Parameter is 0 while for OGR between these extremes the value gradually

increases to a maximum before returning again to 0. The maximum value for the Error Parameter at 3415.69 psia is about 1.017. The positive error indicates that the  $OGR_{PSM}$  is higher than the  $OGR_{BO}$  and the black oil model gives conservative oil rates. The maximum errors for 2500 psia and 1500 psia are 1.028 and 1.048 respectively.

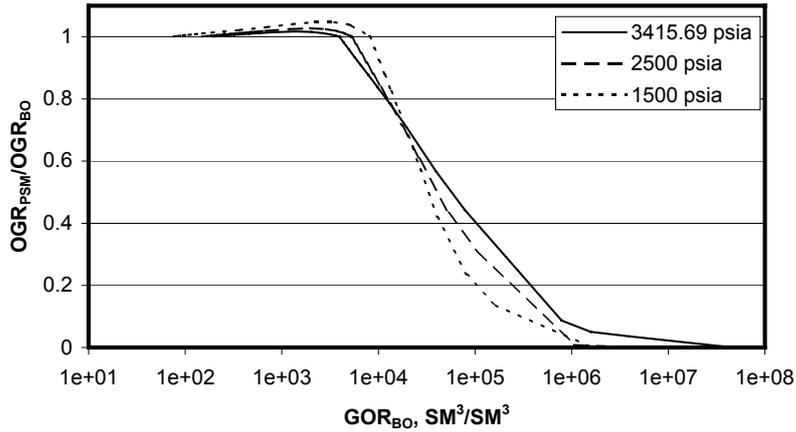


**Fig. 2.33 – Error Parameter  $OGR_{PSM}/OGR_{BO}$  plotted as a function of  $OGR_{BO}$  for 3 different pressures.**

Rovere et. al<sup>13</sup> have reported similar observations. They however plot  $q_{oP}/(q_{oo}+q_{og})$  vs.  $GOR_{BO}$ . The parameter on the y-axis is essentially our Error Parameter  $OGR_{PSM}/OGR_{BO}$ . When we plot vs.  $GOR_{BO}$ , Fig. 2.34 matches their plot.

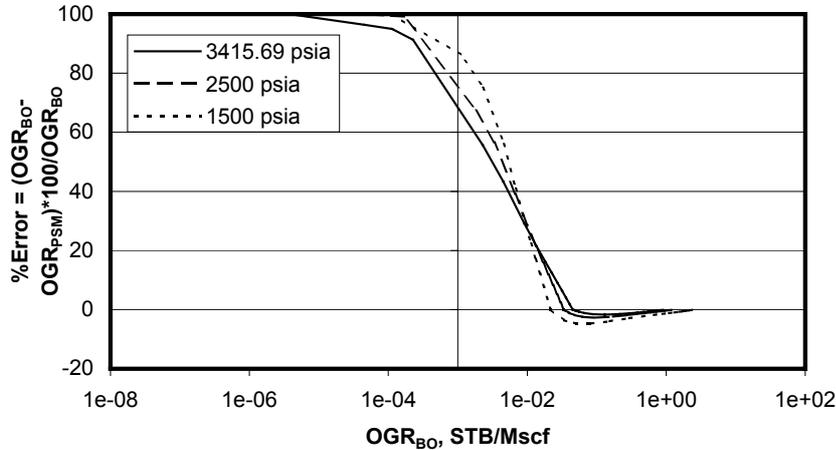


**Fig. 2.34 – Error Parameter  $OGR_{PSM}/OGR_{BO}$  plotted as a function of  $GOR_{BO}$  for 3 different pressures.**



**Fig. 2.35 – Error Parameter  $OGR_{PSM}/OGR_{BO}$  plotted as a function of  $GOR_{BO}$  for 3 different pressures and for mixtures of EO+EG and EG+IG.**

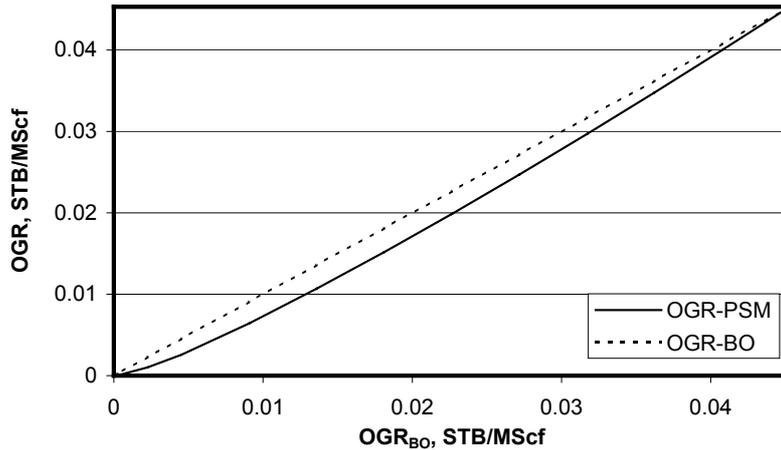
While they limit their observations to mixtures of EO and EG, the injection gas implementation in the BOz method prompts us to also investigate errors in mixtures of EG and IG. Instead of a separate plot we extend Fig. 2.34 to include mixture of EG and IG. The low values of OGR for the mixtures, terminating with an OGR of 0 for the IG (or a GOR of infinity), necessitates a log scale for the x-axis. This plot (Fig. 2.35) shows that the Error Parameter goes down steadily from 1 to 0. The BO model in this case over-predicts the oil production. Substantial errors are observed as the EG becomes leaner and leaner.



**Fig. 2.36 – % Error in surface oil production relative to BO amounts as a function of  $OGR_{BO}$  for 3 different pressures. The thin solid line represents an oil yield of 1 Barrel per Million.**

To obtain a measure of the magnitude of errors, we define  $\%Error = (OGR_{BO} - OGR_{PSM}) * 100 / OGR_{BO}$ . The plot (Fig. 2.36) of this  $\%Error$  vs.  $OGR_{BO}$  shows that at an OGR of 1 barrel per million (.001 STB/Mscf) the error in oil rates could be over-estimated by 70-90%. But the small oil rates, and the fact that they occur late in life, translates to negligible errors in recovery.

A plot (Fig. 2.37) of the functions themselves, for which the  $\%$  error is being calculated, shows that while one ( $OGR_{BO}$ ) is linear, the other is not. Both go towards zero, but at different rates.



**Fig. 2.37 – $OGR_{BO}$  and  $OGR_{PSM}$  functions showing the difference in approach to zero as OGR tends to zero**

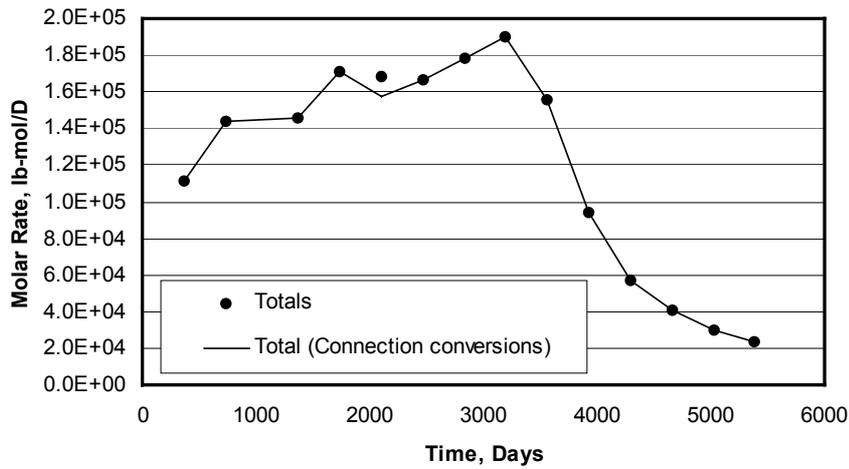
#### Reasons for the difference in $OGR_{BO}$ & $OGR_{PSM}$

As explained previously, the  $OGR_{BO}$  comes from separate processing of reservoir oil and reservoir gas through the surface process (indirectly through use of the ECL PVT tables). If a flowing mixture contains 50% IG and 50% EG, the  $OGR_{BO}$  will be the 50-50 recombination of the OGRs for IG ( $= 0$ ) and EG ( $= R_v$ ), as if they are being processed independent of each other. This gives a linear relationship between the OGR based mixing and the resulting OGR. But when the mixture (combined 50-50 on OGR basis) results in a wellstream (after conversion through PSM), the mixture goes through the process together, affecting each other. Adding IG to EG causes the process to extract less of the oil in EG resulting in less oil recovery than the BO gives us. The leaner the mixture is (containing less and less of EG), the less efficient the process is. So the  $\%$  error goes up. In the limit of an epsilon amount of EG in IG ( $OGR \rightarrow 0$ ) almost all of the oil goes into the gas giving a 100% error. For pure IG, of course, there is no oil so the  $\%$  error is zero.

### 2.3.5 Why BOz Conversions are Necessary on a Connection Basis?

The BOz method converts black-oil surface rates to wellstream molar rates. For greatest accuracy, the conversion is performed on a grid-block connection basis. Most detailed information from the simulator is available at this level.

For a given producing well (or field) OGR, a unique flowing well composition may not exist. For simplicity, consider a well with two connections. One connection produces a mixture of the original reservoir gas and the injection gas, whereas the other connection produces reservoir oil only. For arguments sake, let's say the total well producing OGR happens to equal the same value as  $R_v$  for the equilibrium gas.



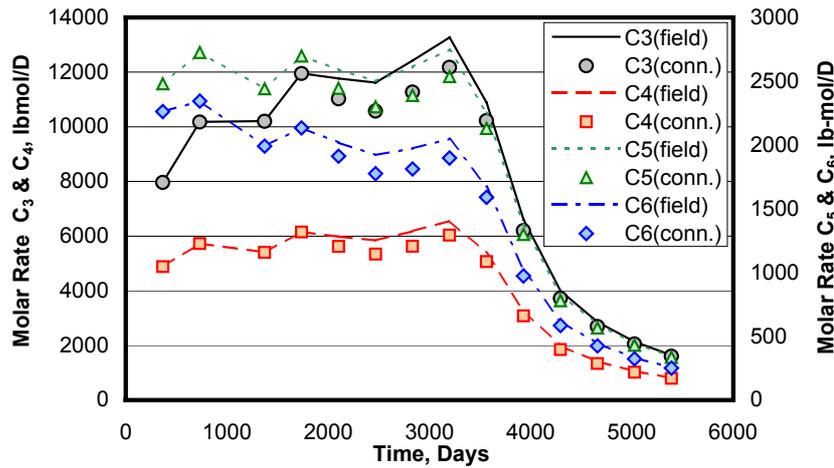
**Fig. 2.38 – Near identical "Total" molar rates for both field based and connection based PSM conversions**

Performing the calculations on a total well basis would result in a flowing well composition equal to the equilibrium gas composition. The actual wellstream composition will be quite different from this erroneous estimate, being some average of injection gas, equilibrium reservoir gas, and equilibrium reservoir oil. With calculations made on a connection basis (in this example), the wellstream would be almost exactly correct. By performing the calculations on a connection basis the best possible wellstream estimate is guaranteed (given the approximation of a black-oil model).

For reservoirs with vertical communication and a constant initial composition produced by depletion (or water injection), the conversions may be performed on a well basis. However, it is recommended to perform the conversions on a connection basis to be sure that the results are as accurate as possible.

### Conversion on field basis versus connections basis

This section uses data from the BOz conversion in a field<sup>a</sup> with 6 production and 1 gas-injection wells. A comparison of field molar rates obtained from a BOz conversion based on *field* oil & gas rates (FOPR and FGPR respectively in ECL100) was made with those from a conversion using connection oil & gas rates (COFR and CGFR respectively) later summed to give field molar rates. While the plot of **total** molar rates (sum of individual components) gives practically identical results (Fig. 2.38), plots of individual component molar rates show up the differences.



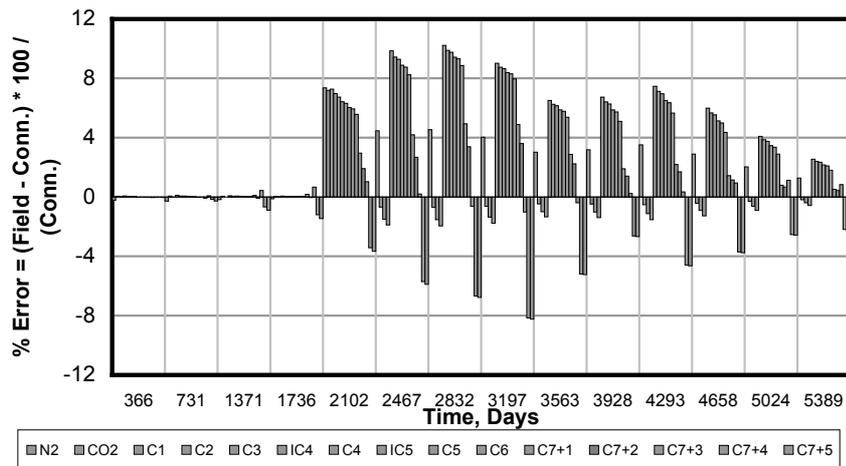
**Fig. 2.39 – Marked differences in molar rates for components important for LPG plant (C3 to C6)**

The signs of errors in component molar rates causes the errors to get practically cancelled to gives the erroneously similar results in the Total Molar Rates plot. Similar plots for C3 to C6 however show the differences markedly in Fig. 2.39. These are the components contributing most to the LPG recovery plant and an average difference of about 7% between the 6<sup>th</sup> and 14<sup>th</sup> years (in this case the field basis over-predicting the recovery) has a major effect on the design of surface process facilities.

A component-wise plot of percent error (Fig. 2.40) summarizes the potential errors from BOz conversion on field rates. Whereas it starts at almost zero initially when the fluid types flowing in different connections are very similar to each other and to the initial single-phase fluid. The slight divergence

<sup>a</sup> Confidential company example. Gas Cap with underlying oil. Gas injection in gas cap with injection well converting into a producer later.

towards the start of 6 years is due to the difference in the degree of 2-phase flow in the connections, each contributing to the rise in % error. Around that time the field experiences gas break-through and the % error increases dramatically. At time 1736 days the field OGR is 0.1175 STB/Mscf, whereas the connection OGRs vary from 1.3493 to 0.0504. This means that while some of the connections flow single phase oil and almost single phase gas, and others flow a mixture, the field rates indicate that the whole field is flowing a particular mixture corresponding to the OGR of 0.1175 STB/Mscf. The differences in compositions for this and the next time step are shown in Table 2.8. Also shown are the compositions at 2832 days, which corresponds to the biggest % errors in Fig. 2.40. These differences are apparent in the compositions too.



**Fig. 2.40 – Component wise % error summarizes the potential errors from BOz conversion on field rates.**

**Table 2.8 – Molar rates for field- and connection-based BOz conversions for 3 different time steps**

	1736 days		2102 days		2832 days	
	Field	Connection	Field	Connection	Field	Connection
N2	1.2079	1.2095	1.2403	1.2334	1.2942	1.2387
CO2	8.1444	8.1426	8.2528	8.2203	8.4486	8.5131
C1	61.7422	61.7506	62.8613	62.5608	64.8185	65.8628
C2	8.5407	8.5365	8.5835	8.5666	8.6754	8.8527
C3	6.9929	6.9910	6.9808	6.9824	6.9736	6.3298
IC4	1.5549	1.5545	1.5363	1.5411	1.5077	1.3728
C4	2.0436	2.0431	2.0121	2.0206	1.9622	1.7887
IC5	0.6876	0.6875	0.6707	0.6753	0.6423	0.5873
C5	0.8912	0.8910	0.8664	0.8731	0.8245	0.7546
C6	1.2488	1.2483	1.1977	1.2112	1.1075	1.0179
C7(1)	1.6834	1.6807	1.4708	1.5251	1.0863	1.0357

C7(2)	2.4892	2.4905	2.1278	2.2290	1.4689	1.4216
C7(3)	1.8160	1.8044	1.4515	1.5338	0.7925	0.7979
C7(4)	0.7381	0.7472	0.5756	0.6363	0.3054	0.3275
C7(5)	0.2194	0.2227	0.1723	0.1910	0.0923	0.0990

### Conversion on well basis versus connections basis

To check if these differences are apparent when the conversion is done on a well basis, a time step of 2832 days (corresponding to the biggest % errors) was chosen and BOz conversion done using the well oil and gas rates (WOPR and WGPR). The pressure used was the corresponding well bottom hole flowing pressure (WBHP). These were compared with connection based molar rates after summing them for respective wells. A table of percent error is given in Table 2.9. It shows that the error in performing the BOz conversion on well basis for this case is negligible for all wells except Well#6, which is an injection well converted into a producer. Also, in general the error is relatively higher for C<sub>7+</sub> fractions. In fact the % error for the last two fractions for this well is 49% and 99%, although the amounts of these themselves is negligible for this well which produces essentially injection gas.

**Table 2.9 – Percent error when BOz conversions are done on a well basis (compared to connection basis conversions) at the time 2832 days.**

Well#1	Well#2	Well#3	Well#4	Well#5	Well#6	Well#7
-0.0007	0.0068	0.0077	0.0096	0.0056	-0.4122	-0.0331
0.0077	0.0037	0.0007	0.0062	0.0124	0.0622	0.0220
0.0095	0.0023	0.0096	0.0114	0.0112	0.1331	0.0070
-0.0005	-0.0031	-0.0066	-0.0017	0.0006	0.1609	0.0266
-0.0098	-0.0057	-0.0082	-0.0141	-0.0251	<b>-1.1393</b>	-0.0026
-0.0178	-0.0146	-0.0177	-0.0362	-0.0560	<b>-1.1353</b>	-0.0255
-0.0206	-0.0141	-0.0197	-0.0398	-0.0680	<b>-1.1329</b>	-0.0370
-0.0199	-0.0131	-0.0191	-0.0451	-0.0959	<b>-1.1197</b>	-0.0735
-0.0245	-0.0165	-0.0182	-0.0517	-0.1024	<b>-1.1146</b>	-0.0777
-0.0095	-0.0045	-0.0064	-0.0440	-0.1270	<b>-1.0888</b>	-0.1028
0.0680	0.0512	0.0622	0.0340	-0.0757	-0.7770	-0.0546
0.0726	0.0606	0.0721	0.0765	0.0264	-0.5855	0.0075
0.2514	0.2305	0.2767	0.3030	0.1892	<b>1.3072</b>	0.0981
-0.6432	-0.6965	-0.8274	-0.7475	-0.3647	<b>49.4118</b>	-0.1861
-0.6669	-0.7199	-0.8553	-0.7754	-0.3693	<b>99.8374</b>	-0.1880

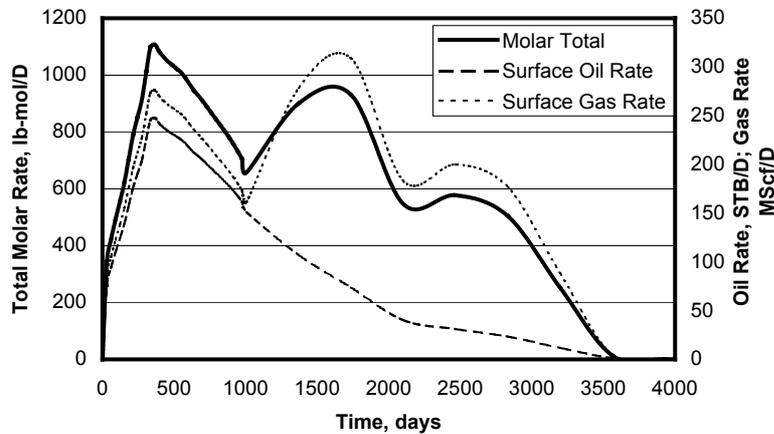
An analysis of the well OGRs and the OGRs obtained by sending the PSM computed well compositions (totaled from connections) through a separator test, shows that the % difference is about 3% for all wells except the injection well. This, plus the fact that all the connections for each well are producing very similar fluids and at similar pressures, ensures that the results for well-based conversions are so close to the connection based ones.

### 2.3.6 Quality Checks on BOz results

The general trend (qualitatively) of total component rates (vs. time) should follow the “combined” trend of surface oil and surface gas for any connection (Fig. 2.41) and also on well (or field) basis. In a sense the program converts a combination of surface oil and gas rates into molar rates.

For an under-saturated reservoir flowing single phase oil or gas, the converted molar rates should give a composition identical to the original composition of the feed (in the PVT program used) which was used to define the ECL PVT tables.

For any connection, the converted molar rate at any time step, when fed through the same surface separation scheme as defined during calculation of ECL PVT tables (e.g. in PVTx), should result in the same oil and gas volumetric rates as those on which the conversion was based in the first place. This was done by two independent methods:



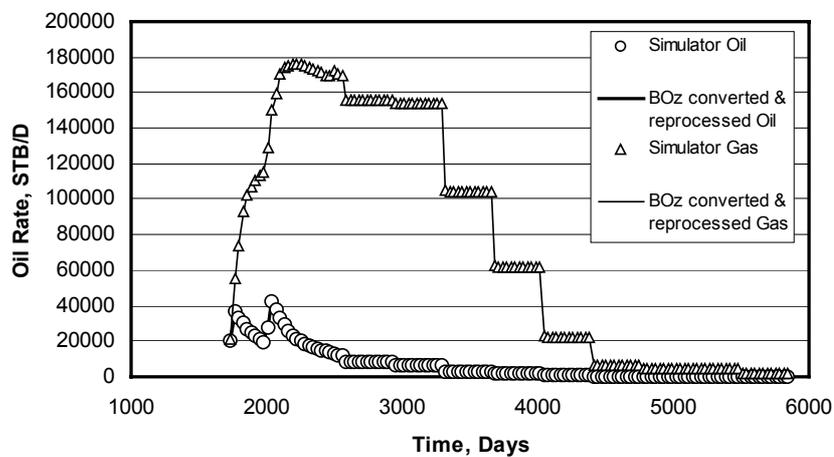
**Fig. 2.41 – A qualitative check on BOz conversion is obtained by plotting the trend of Molar rates vs. “combined” trend of oil & gas rates.**

The composition resulting from the molar rates was used as feed and flashed to conditions of the 1<sup>st</sup> stage separator. The resulting oil composition and vapor mole fraction ( $\beta_1$ ) were noted. This composition was used as flashed to the conditions of the 2<sup>nd</sup> stage separator and the resulting oil composition and vapor mole fraction ( $\beta_2$ ) were obtained. Finally this composition was flashed to the conditions of the 3<sup>rd</sup> stage separator to get resulting oil composition and vapor mole fraction ( $\beta_3$ ). The density ( $\rho_o$ ) and Molecular Weight ( $M_o$ ) of this stock-tank oil gives the volume, per mole of original feed, as:

$$V_o = (1 - \beta_1)(1 - \beta_2)(1 - \beta_3) \frac{M_o}{\rho_o} \dots\dots\dots (2.73)$$

This was converted to actual STB/D for the molar rate (lb-mol/D) used. The percentage difference between the original and final volumetric rates was always less than half a percent for connection rates tested.

An identical surface separation system was set up in a commercial process simulator (Hysys<sup>a</sup>), to simulate what was done manually in PVTx. Identical EOS parameters were used as far as possible. Due to lack of possibility of manually specifying the volume shift parameter in Hysys, some variation in density and volumes were expected. The molar rates were fed into the Process and results (Surface oil in STB/D and surface Gas in Mscf/D) are immediately given as output. Although the % difference is higher (1-5 %), the similarity in results was a useful tool to check the BOz conversions very quickly.



**Fig. 2.42 – A quantitative check on BOz conversion is obtained by the original Simulator black oil rates vs. BOz converted molar rates reprocessed through an identical surface process.**

At this point is worth mentioning that Streamz now has an in-built facility to perform this check using the **Process** command. This is a regular check applied to any BOz conversion from a black-oil simulator. Such a back check is reported in Fig. 2.42.

<sup>a</sup> Hysys is a commercial process simulator for the MS Windows operating platform. It conforms to Object Linking and Embedding (OLE) and automation standards for this operating system allowing it to be started and controlled from another similarly complaint software. This allowed the partially automated re-processing of PSM converted molar rates by controlling Hysys from MS Excel.

### 2.3.7 Negative Molar Rates

When the BOz method is applied to BO simulator rates at very special conditions, they stretch the method to its limits. As has been shown, this method assumes the flow of either equilibrium gas, equilibrium oil or a mixture of these two fluids. Special extension covers the case of gas injection when a mixture of equilibrium gas and injection gas is assumed to flow.

When the simulator conditions result in the flow of a fluid very close to, or exactly, the equilibrium phase, the BOz method is operating at its limits. At these conditions the negative split factors of certain components start to dominate. Such negative split factors usually occur for the heaviest *gas* components or the *lightest* oil components. If the negative split factor for a gas component is slightly *too much* negative than it should be, for reasons we will see later, it may not be sufficiently compensated by the positive oil component split factor. In such cases the weighted sum,

$$m_i = q_o S_{oi} + q_g S_{gi} \dots\dots\dots(2.74)$$

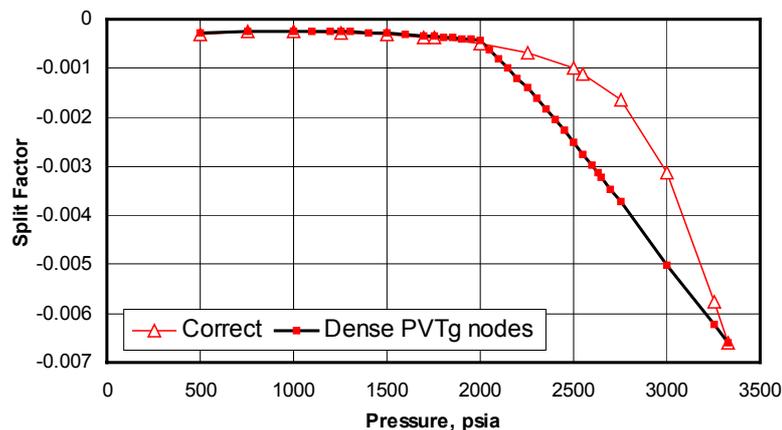
becomes negative. Typically in such cases the component molar rate is a very small number (practically zero) but may end up being a very small *negative* number. For practical application of this method a negative amount of component molar rate cannot be tolerated. In this section we look at sources of this problem and arrive at some guidelines to prevent them.

#### Interpolation

The BOz method is implemented as a series of split factors based on tables (Extended Black-Oil PVT data) generated at the same time a PVT program generates BO PVT tables for a black oil reservoir simulator. These split factors are thus valid 100% at the pressure nodes used in the depletion-type experiment used. When the reservoir simulator calculates the surface oil and gas production rates it uses the BO PVT tables supplied. If the current grid cell pressure does not match the node pressures exactly it uses linear interpolation between the closest available nodes.

The BOz split factor interpolation in Streamz is linear. This usually takes care of most situations. In certain cases, however, the node pressures are necessarily “too” sparse. If the PVT properties generated by the depletion experiment are non-monotonic the simulator does not accept them. The normal solution for this is to re-generate the BO PVT table by excluding those pressure nodes in the depletion experiment. If the split factors were kept consistent, they too would be re-generated based on the simultaneously generated extended BO table. The result is a sparse split factor table (at least in some parts).

Another subtle issue involves the use of PSM's **Boz2Cnv** module and splicing of Oil and Gas PVT tables. It is a common practice to use oil tables from one experiment (typically DLE) and gas tables from another experiment (typically CVD). It is also common that the depletion pressure nodes in each experiment are not identical. To generate consistent split factor tables, splicing is also done while generating them. The generation of split factors always linearly interpolates the contents of the extended BO PVT table for each intervening pressure node in each table being spliced.



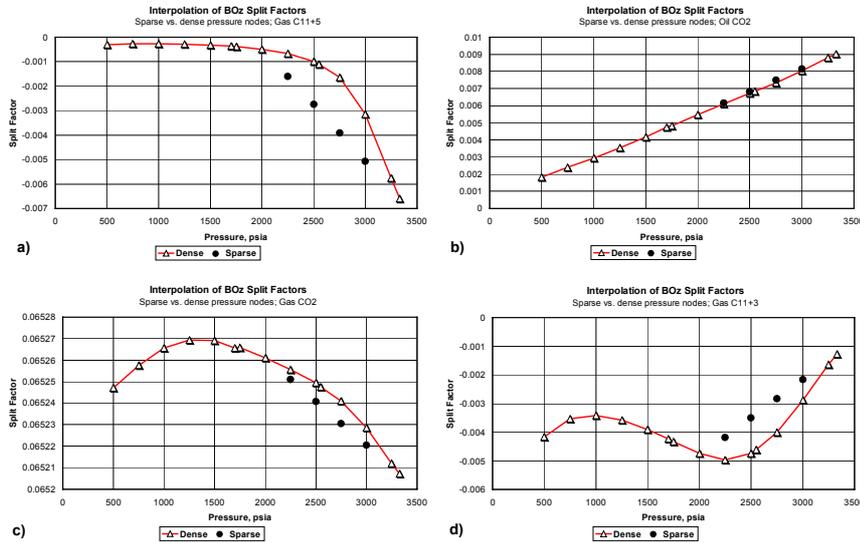
**Fig. 2.43 – Pseudo-dense Gas split factors shown together with the correct ones for component C11+5.**

This generates a *pseudo*-dense split factor tables with apparent dense pressure nodes as shown in Fig. 2.43. At any pressure between 2000 psia and the initial saturation pressure (approx. 3400 psia) a “too negative” value of the gas split factor will be used for this component resulting in a potential for negative molar rates. If the overall OGR is close to the equilibrium  $R_v$  at that pressure (the highest possible gas rates) there is a big chance of negative molar rates.

In such cases it is immaterial whether the split factors have sparse or pseudo-dense nodes, the effect is exactly the same. The only difference is that in the latter case the user mistakenly assumes that the nodes are dense and hence concludes that the reason for the negative molar rates is not interpolation.

Fig. 2.44 shows some of various shapes of split factor vs. pressure curves. Fig. 2.44a) is the same as the one discussed earlier with a big potential for negative molar rates. Fig. 2.44b) shows an essentially linear relationship and any amount of sparseness would still give “correct” results. Fig. 2.44c) is also very non-linear with the interpolation at points of interest (2000 psia to 3400 psia) giving too small values. But since the value of the split factors for this component is positive, it does not give too *negative* and thus has no potential for negative

molar rates. Fig. 2.44d) is also very non-linear and also the values are in the negative range. But at the points of interest the curve increases with pressure and the interpolation would result in too high negative value. While the actual error still exists, it does not result in negative molar rates and would be more acceptable. It must be noted that, for the components for which this is a potential problem, the amounts are extremely small and so the main objective is to make them physically acceptable (i.e. positive).



**Fig. 2.44 – Various shapes of Split factor vs. pressure curves. When interpolated at points shown by solid circles, a) results in too negative, b) has no effect, c) results in too small, but positive, values, and d) results in smaller negative values. Only a) has potential for negative molar rates.**

The conclusion from the above discussion is that the original split factor tables should be as dense as possible to avoid interpolation and a potential for negative molar rates.

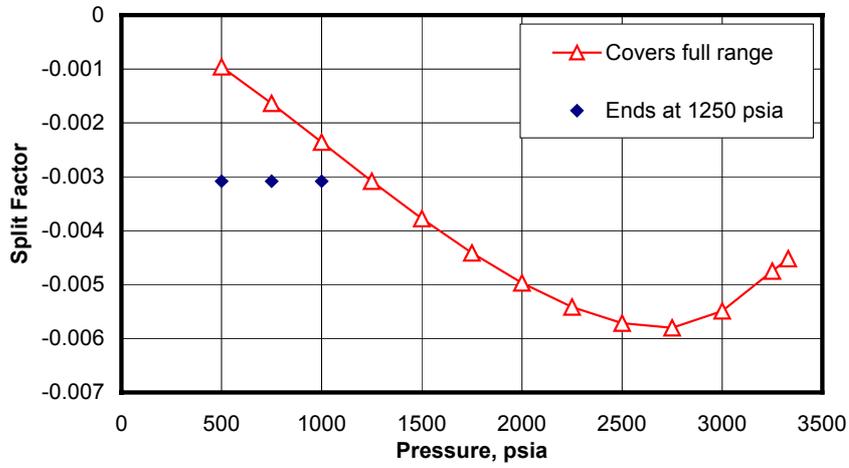
### Extrapolation

PSM Split factors are generated at specified nodes and are interpolated for pressures within those nodes. If the pressure extends beyond the limiting nodes PSM uses constant extrapolation. In such situations some combination of split factor shape and value may result in “too” negatives and hence a potential for negative molar rates. Such an example is Fig. 2.45.

### Limiting situation for negative molar rates

The limiting situation can be determined easily by expanding Eq. (2.74). To prevent negative molar rates for any component we have the following equation.

$$m_i = q_o S_{oi} + q_g S_{gi} > 0 \dots\dots\dots(2.75)$$



**Fig. 2.45 – “Too negative” split factors due to extrapolation causes a potential for negative molar rates. This is possible if the full range of pressures are not covered by the split factor tables.**

$$q_o S_{oi} > -q_g S_{gi} \dots\dots\dots(2.76)$$

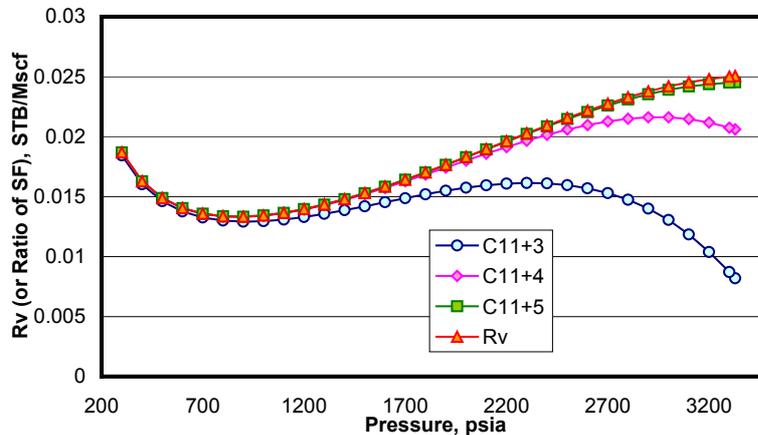
$$\frac{q_o}{q_g} > -\frac{S_{gi}}{S_{oi}} \dots\dots\dots(2.77)$$

$$OGR > -\frac{S_{gi}}{S_{oi}} \dots\dots\dots(2.78)$$

Thus we have a very simple criterion for determining whether the producing fluids from a reservoir simulator would result in negative molar rate. The smallest value of OGR occurs when only the equilibrium gas is flowing and  $OGR = R_v$  and it is at this point that the biggest potential for it becoming lower than the ratio  $S_{gi}/S_{oi}$  exists. A similar criterion can be developed for a fluid producing at close to equilibrium oil with  $GOR = R_s$ .

$$GOR < -\frac{S_{oi}}{S_{gi}} \dots\dots\dots(2.79)$$

These criteria are only applicable for components that have a negative component split factor (either gas or oil) while the other (oil or gas) component split factor is positive. The criterion given by Eq. (2.78) is always applied to the heaviest components when the producing fluid is close to the equilibrium gas. The criterion given by Eq. (2.79) is applied to the lightest components when the producing fluid is close to the equilibrium oil.



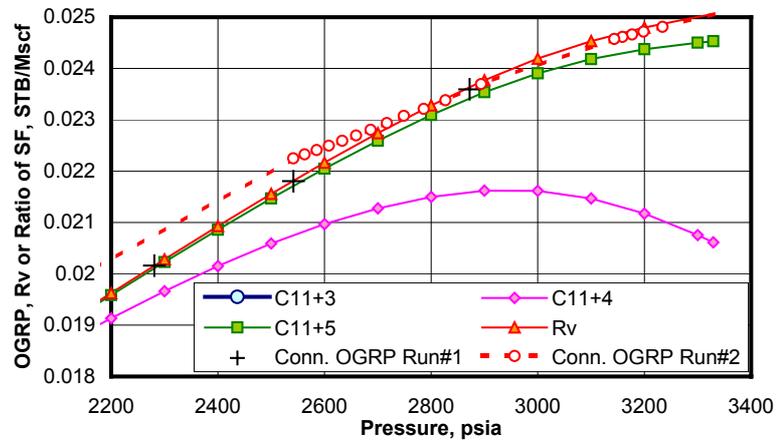
**Fig. 2.46 –  $R_v$  & Ratio of Split Factors vs. Pressure for the last 3 components C11+3, C11+4, and C11+5.  $R_v$  is the limiting producing OGR that ensures positive molar rates.**

Fig. 2.46 shows this criterion graphically for the last three components of the test fluid  $C_{11+3}$ ,  $C_{11+4}$ , and  $C_{11+5}$ . If the producing OGR, for some reason, goes below the ratio of split factors for any of these components then the calculated molar rate for that particular component will turn up negative. From Fig. 2.46 it is clear that at higher pressures this is most likely for only component  $C_{11+5}$ . One situation would be if sparse BO PVT tables have been supplied to the simulator that is producing essentially equilibrium gas. It interpolates a value at a pressure of 2700 psia that ends up being less than the ratio of split factors for  $C_{11+5}$  at that pressure.

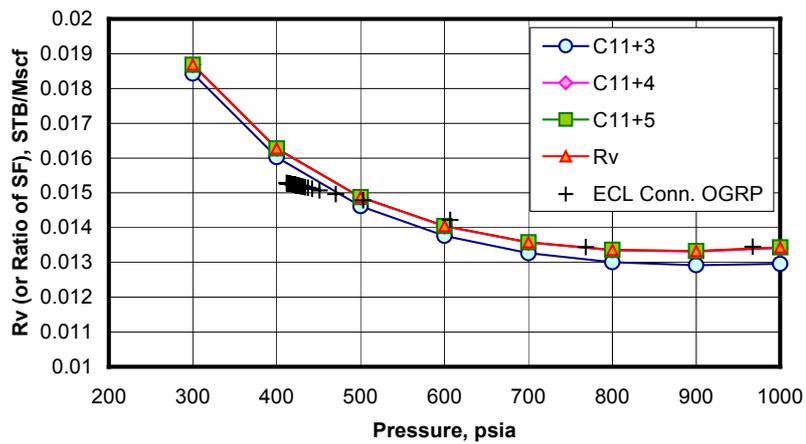
Other likely situation when this could be true is when the simulator extrapolates the  $R_v$  table towards the low-pressure end and reports an OGR less than the true  $R_v$ . Since all 3 components are very close to the limiting  $R_v$  value at low pressures, all three would end up with negative molar rates.

Fig. 2.47 depicts a zoomed portion of the OGRP from two separate ECL100 runs plotted on the  $R_v$  and  $S_{ig}/S_{io}$  vs. pressure plot. Both runs use the same BO PVT tables consistent with the  $R_v$  plotted. Connection OGRP for Run#1 (+ symbols) show that this simulator was producing exactly the equilibrium

gas. Although this represents the worst possible situation, the slight higher values than the values of  $S_{ig}/S_{io}$  at these high pressures ensure that no negative molar rates result.



**Fig. 2.47 – OGRPs from two separate simulator runs, both using same BO PVT, plotted alongside  $R_v$  and ratio of component split factors. C11+3 plot is not visible in this zoomed range.**



**Fig. 2.48 – Low-pressure range plot of OGRP from a simulator run, along with  $R_v$  and ratio of component split factors. Simulator extrapolation (below 768 psia) could result in negative molar rates for pressures below 500 psia.**

Connection OGRP for Run#2 (dashed lines with circles) show that the simulator produces fluid quite close to the equilibrium gas. At some pressures the interpolation within the simulator results in OGRP lesser than the equilibrium gas  $R_v$  value at those pressures. This would normally result in

negative molar rates at lower pressures, but at these higher pressures the divergence between the values of  $R_v$  and  $S_{ig}/S_{io}$  avoid any negative as the criterion is still honored. The curve for  $C_{7+3}$  is not visible in this zoomed plot.

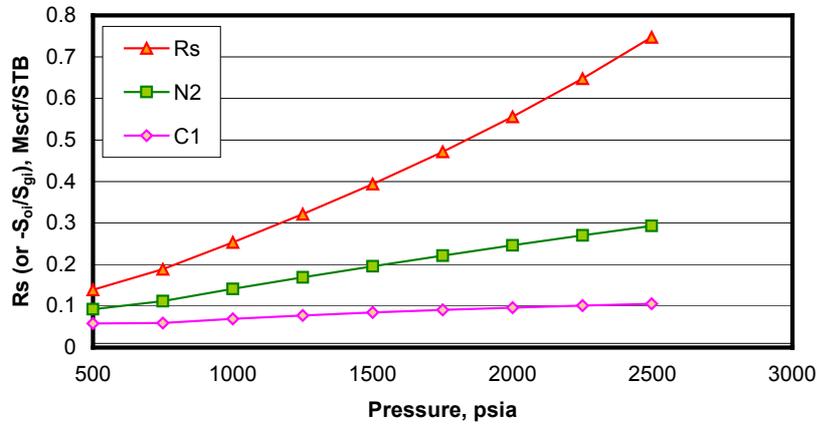


Fig. 2.49 – Plot of the ratio of oil and gas split factors with that of saturation gas-oil ratio,  $R_s$ , as the limiting producing GOR.

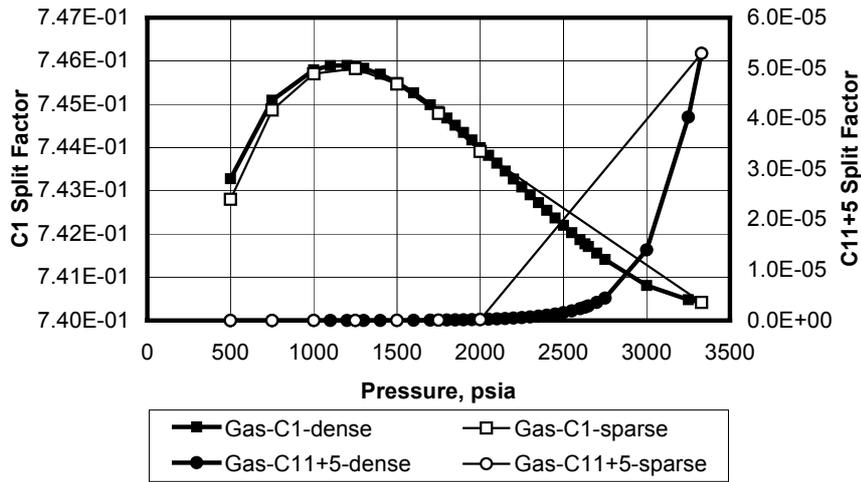


Fig. 2.50 – Plot showing the slight differences in values of split factors due to the difference in the number of depletion pressure nodes in the depletion-type experiment used to generate the split factors.

Fig. 2.48 depicts a zoomed low-pressure portion of the OGRP from an ECL100 run plotted on the  $R_v$  and  $S_{ig}/S_{io}$  vs. pressure plot. Connection OGRP (+ symbols) shows that extrapolation within the simulator occurs below the pressure 768 psia. This results in OGRP lesser than the equilibrium

gas  $R_v$  for pressures below 500 psia. Since the criterion is violated this results in negative molar rates. This is a typical situation because the Eclipse 100 simulator always performs a linear extrapolation using the last two table entries. If the table supplied does not extend to cover the producing range, and if the split factors are not monotonic at such low pressures, the producing (extrapolated) OGR is likely to produce negative molar rates for the three  $C_{11+}$  components.

To avoid negative molar rates in a non-gas injection case the BOz method ensures that we only operate in the saturated region. If the simulator reports an OGRP smaller than the saturated  $R_v$  at that pressure, BOz logic will supply the saturation pressure of the reported OGRP for use in the BOz split factor tables. This is equivalent to pulling back the point for operation within the saturated region.

For case of gas injection, of course, separate split factor tables are used which is consistent with an  $R_v$  value between saturated  $R_v$  at that pressure and zero.

Fig. 2.49 depicts the other criterion of the limiting producing GOR. From the large differences between it and the ratio of  $S_{io}$  and  $S_{ig}$  even at very low pressures it is clear that the simulator is very unlikely to cross this boundary.

#### Other sources of differences

We have seen that sparseness in the PVT tables and the split factors are sources of negative molar rates. Although this section discusses the sources and avoidance of negative molar rates, we highlight a subtle point in production of sparse tables. We have seen that at times sparse tables are a necessity. One way to produce sparse tables is to have sparse nodes in the depletion-type experiment(s) used in the PVT program. While using sparse BO PVT tables we have seen that we need to use dense split factors to avoid negative molar rates. One way to produce dense split factor tables is to re-run the PVT program with dense pressure nodes in the depletion-type experiment(s). Fig. 2.50 however shows that slight differences in the values of split factors, at the same pressure, may result from these two depletion-type experiments. The best way to avoid this is to use the dense-most depletion stage nodes, which is also more consistent with the way the simulation works, and then to remove those nodes in the resulting tables as needed (e.g. to avoid non-monotonic tables unacceptable by the simulator).

#### Guidelines to avoid Negative Molar Rates

The following guidelines will ensure positive (and more correct) molar rates when using the BOz method:

- Prevent interpolation / extrapolation as far as possible by ensuring that the PVT program simulating the depletion-type experiment for generation of both simulator PVT tables and BOz tables:
  - a. Cover the **full range** of expected pressures.
  - b. Include as **dense** pressure stages as possible.
- If dense pressure nodes result in non-monotonic Bo and/or Rs tables:
  - a. Remove those nodes in simulator PVT tables.
  - b. Re-introduce those nodes in some consistent manner (e.g. using BOPVT program).

### 2.3.8 BOz Conversion Examples

#### North Sea Full Field case.

A North Sea reservoir modeled with 2 black-oil PVT regions, about 50 wells, and approximately 1000 well-grid connections (well completions) was chosen as an example of a typical field application of the BOz method. Further complications were the existence of gas injection and 2 different surface processes. An estimated 100,000 streams were expected. The conversion was performed on a grid cell connection basis at each time step before averaging & summing to yearly-field streams. The excellent agreement for this complicated case between this conversion and the “correct” rates from Eclipse 300 is shown in Fig. 2.51.

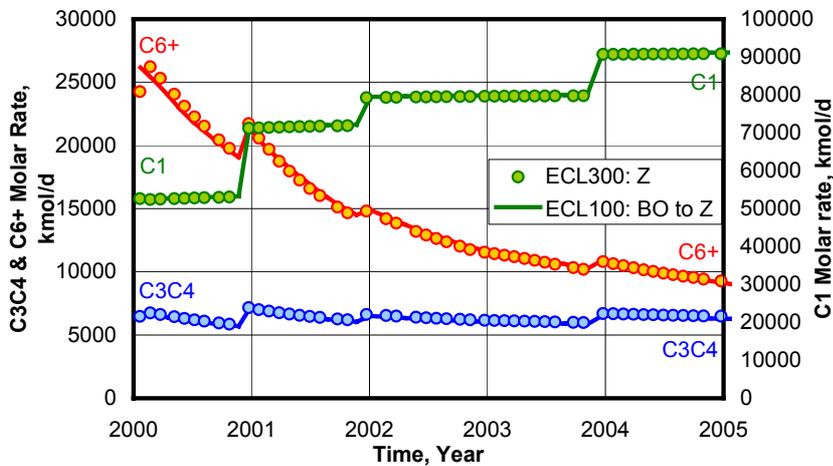
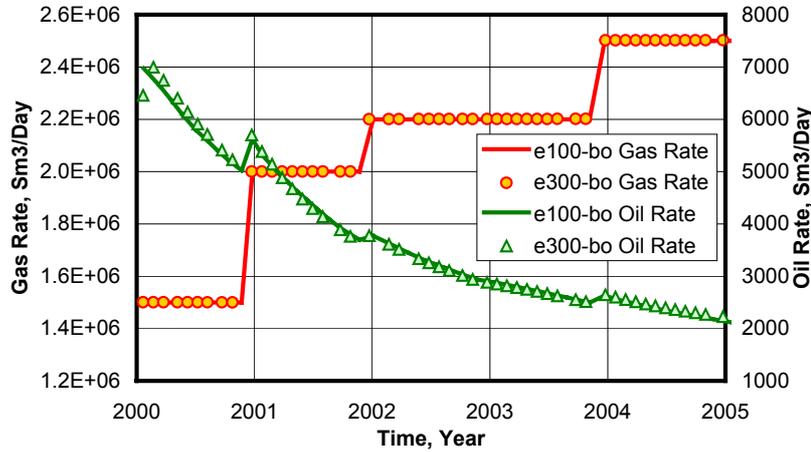
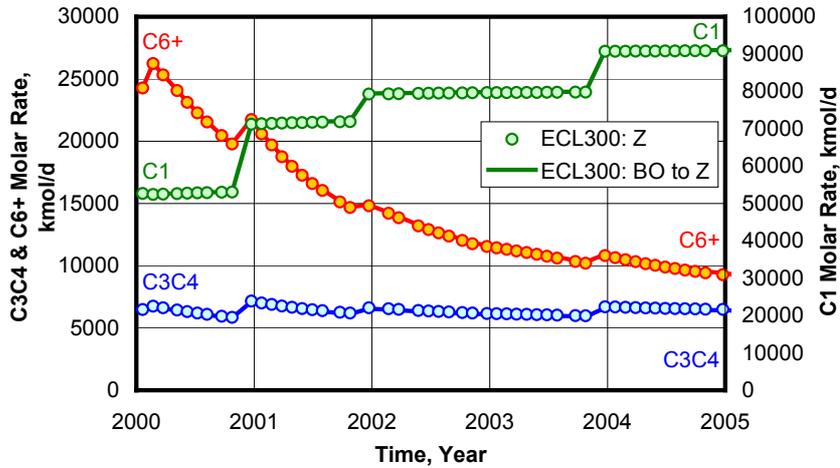


Fig. 2.51 – Comparison of predictions for full-field molar rates from BOz/PSM converted E100 and E300 models.

The slight differences in C<sub>6+</sub> rates immediately following year 2000 seem to be the only differences. BOz conversion is based on the surface oil and gas rates predicted by the model. Normally black-oil based reservoir simulators predict slightly different surface oil and gas rates from a corresponding compositional simulator for the same field. To investigate if the minor differences in Fig. 2.51 are due to the limitation of the BOz method we compare the surface oil and gas rates from the two models in Fig. 2.52.



**Fig. 2.52 – Comparison of predictions for full-field surface oil and gas rates for E100 and E300 models.**



**Fig. 2.53 – Comparison of predictions for full-field surface oil and gas rates for E100 and E300 models using E300 gas & oil rates for BOz conversions.**

Similar differences are observed here too leading to the speculation that the differences in the surface oil and gas rates cause the differences in PSM/BOz

converted molar rates and actual molar rates. This is confirmed in Fig. 2.53 where the perfect match confirms the accuracy of the BOz method.

### CO<sub>2</sub> Immiscible Injection

A North Sea 2D sector model containing a highly under-saturated, low API oil. The reservoir is highly permeable and contains a single horizontal well down structure. CO<sub>2</sub> is injected through an up-dip injector.

This is a very special BOz conversion. Normally the split factors for a gas injection case are prepared based on the assumption that

- a mixture of reservoir oil and reservoir gas flows in the grid-connection when the instantaneous OGR is within  $r_s$  of equilibrium gas and  $1/R_s$  of equilibrium oil, and
- a mixture of reservoir gas and injection gas flows if the instantaneous OGR is within  $r_s$  of injection gas ( $=0$ ) and  $r_s$  of equilibrium gas.

For this BOz conversion the above procedure was modified to assume:

- a mixture of reservoir oil and “CO<sub>2</sub>-swelled oil” flows in the grid-connection when the instantaneous OGR is within  $1/R_s$  of swelled oil and  $1/R_s$  of equilibrium oil.
- a mixture of injection CO<sub>2</sub> gas and “CO<sub>2</sub>-swelled oil” flows if the instantaneous OGR is within  $r_s$  of CO<sub>2</sub> and  $1/R_s$  of “CO<sub>2</sub>-swelled oil”.

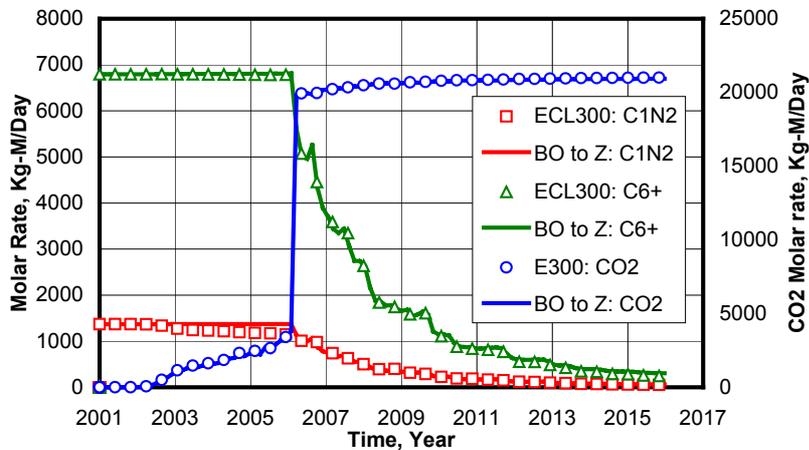
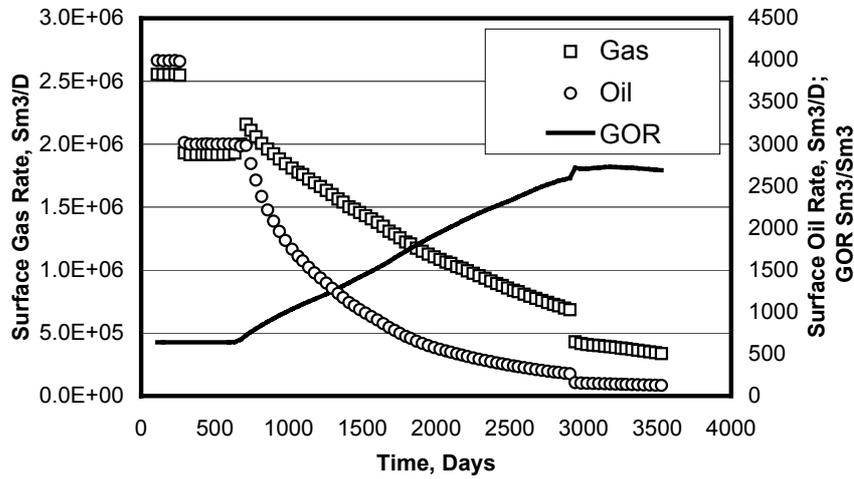


Fig. 2.54 – Comparison of predictions for BOz converted full component molar rates (from E100) and E300 models for the CO<sub>2</sub> immiscible injection case. Courtesy Knut Uleberg<sup>17</sup>.

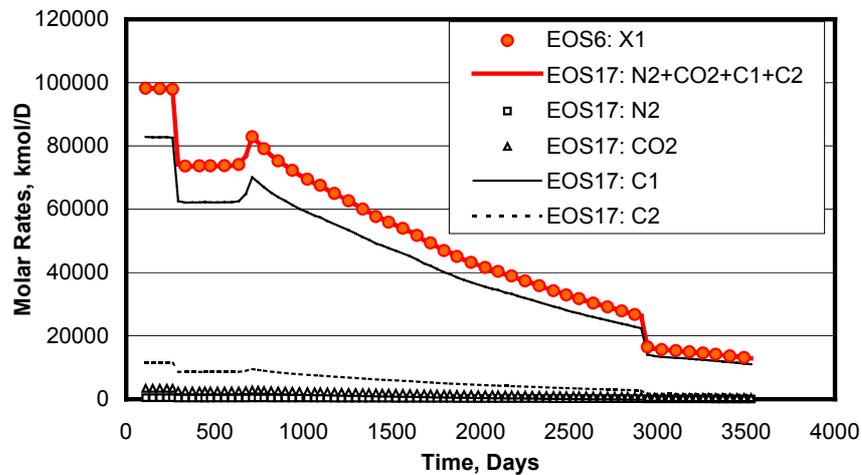
The excellent agreement for this special case between this conversion and the “correct” rates from Eclipse 300 is shown in Fig. 2.54.

**HTHP Rich Gas Condensate Reservoir: Pressure depletion**

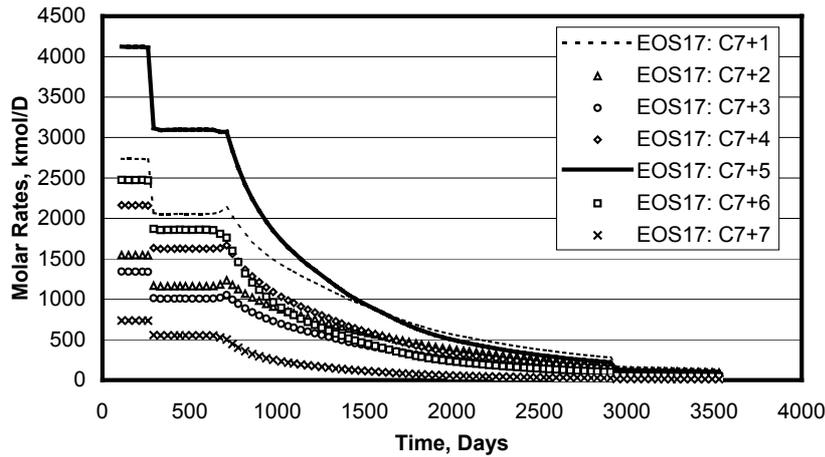
Another North Sea full-field BO model containing a highly under-saturated, rich gas condensate. Two horizontal wells are used to produce this satellite structure.



**Fig. 2.55 – BO reservoir simulator streams that form the basis of conversion to detailed process streams.**



**Fig. 2.56 – Detailed (17c) individual molar rates of the light ends are profiled vs. time starting initially from a BO simulation result. The BOz conversion produces a single component X1 (corresponding to  $N_2+CO_2+C_1+C_2$ ) only.**



**Fig. 2.57 – Detailed individual molar rates of the heavy ends are profiled vs. time starting initially from a BO simulation result.**

The special feature of this case is the double BO→EOS6→EOS17 conversion. This first conversion uses BOz method and the second one used the generic conversion matrix, but again using pressure dependence for the split factors. This example shows the possibility of a consistent conversion from Black-oil simulation results to detailed Process component molar rates where profiles for individual components ( $C_1$  to  $C_6$ ) and heavy ends ( $C_{7+}$ ) are provided for every times step. This detailed information is valued for process simulation for the design of condensate recovery plants and was, in fact, useful for the economic evaluation of this satellite structure.

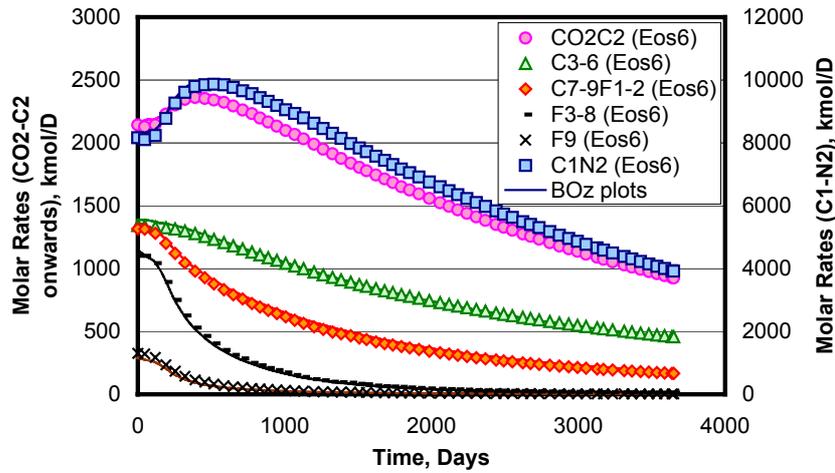
In the absence of equivalent compositional simulation results it is not possible to evaluate the accuracy of the conversions. That has been done with sufficient number of test cases. The purpose her was to demonstrate the utility of a tool to consistently and easily convert BO reservoir simulator streams (Fig. 2.55) to detailed process simulator streams. Fig. 2.56 shows the profile for the light components and Fig. 2.57 for the heavy ends.

### Rich Gas Condensate with Compositional Gradient

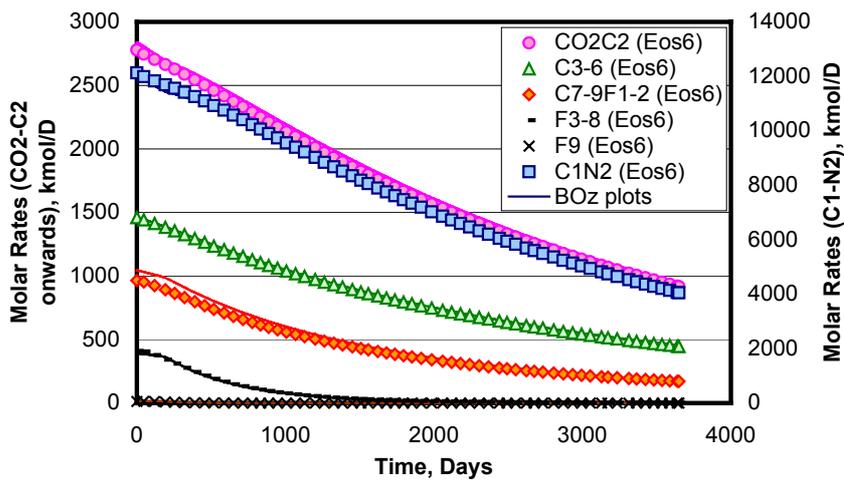
This is from a North Sea 3D Sector model containing near-critical fluid with compositional gradient. 3 producing wells are located in upper-, middle-, and lower structure.

The special feature of this case is compositional gradient resulting in a different initial fluid composition varying with depth. Black oil simulation results are post-processed by the BOz method. Conversions are performed at grid-cell level thereby capturing the correct GOR and converting it to the correct

compositions. Compositions (molar rates) are then summed to well-level for reporting.



**Fig. 2.58 – Fairly accurate calculation of the molar rate profile for the lower structure well.**



**Fig. 2.59 – Very Good matching of molar rates for the middle structure well.**

The special case of compositional gradient is captured fairly accurately. The most dramatic results are from the lower structure well (Fig. 2.58) where the molar rates profiles initially increase for some components before decreasing.

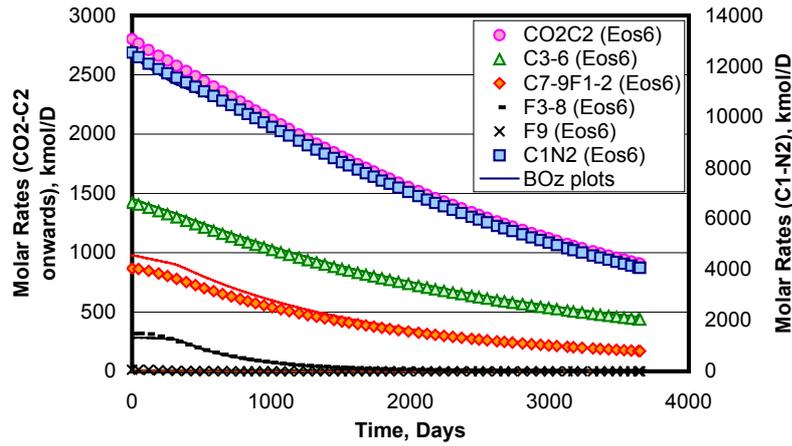


Fig. 2.60 – Very Good matching of molar rates for the upper structure well.

### Gas Injection in Rich Gas Condensate reservoir with constant composition

This is from a North Sea 3D Sector model containing a very rich GC fluid with constant composition. Gas injection takes place from up-dip well and production from down-dip well. Black oil simulation results are post-processed by the BOz method. Conversions are performed at grid-cell level thereby capturing the correct GOR and converting it to the correct compositions. Compositions (molar rates) are then summed to well-level. Conversions from 6-component molar rates are performed at well level, with split factors a function of the well stream  $C_{6+}$  mole fraction, to obtain 22-component molar streams.

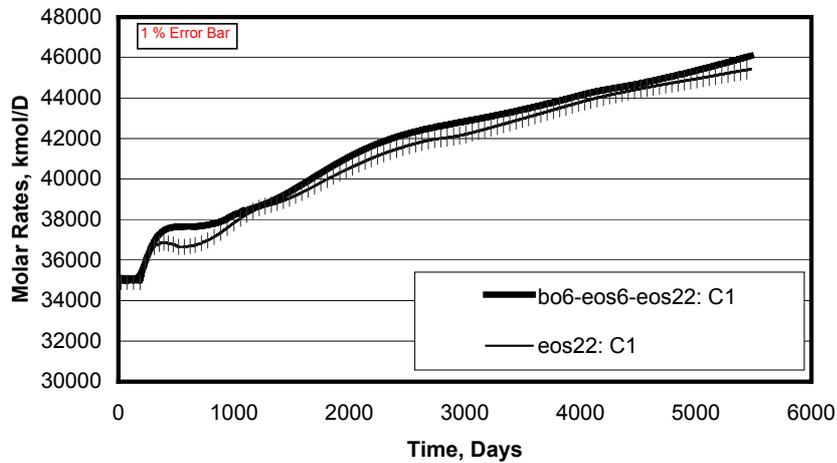
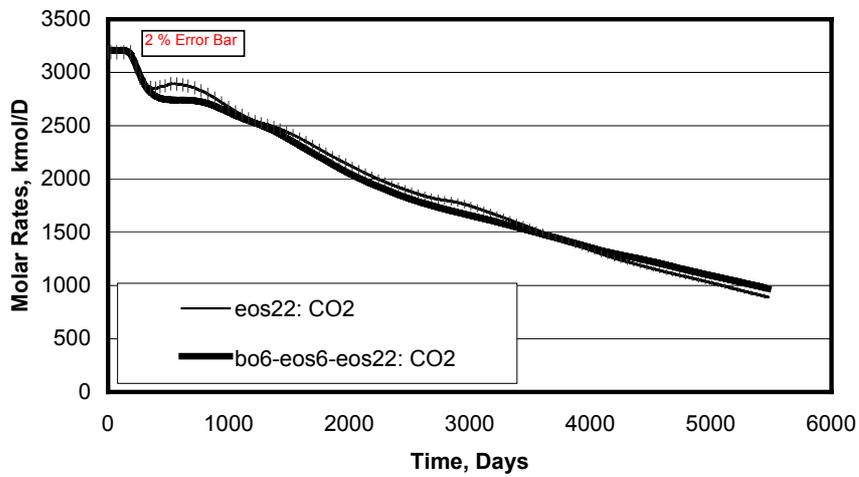


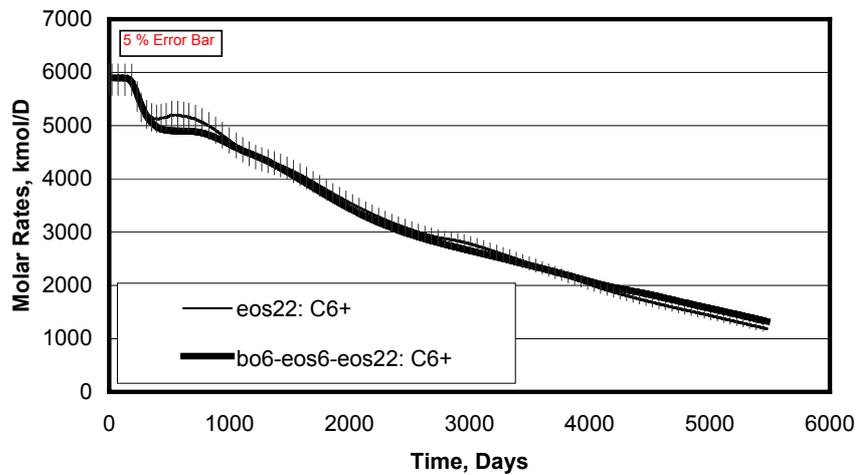
Fig. 2.61 – Excellent double-conversion (bo6-eos6-eos22) match for C1.

The 1% error bars quantify the accuracy of the conversion. The general global trend along with the local undulations are captured quite closely although not to the correct degree (Fig. 2.61).

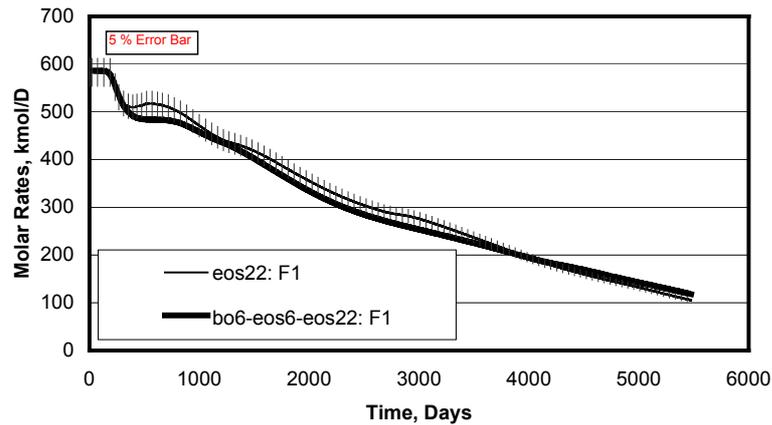


**Fig. 2.62 – Excellent double-conversion (bo6-eos6-eos22) match for CO<sub>2</sub>.**

Fig. 2.62 is a similar comparison for CO<sub>2</sub> where again the differences are within 1% for most part of the simulation. Fig. 2.63 and Fig. 2.64 are for the heavy-ends with the error being within 5%.



**Fig. 2.63 – Extremely good (within 5%) double-conversion (bo6-eos6-eos22) match for C<sub>6+</sub> components.**



**Fig. 2.64 – Extremely good (within 5%) double-conversion (bo6-eos6-eos22) match for the first individual  $C_{6+}$  component F1 of the detailed characterization.**

### 2.3.9 Conclusions

A robust, accurate and efficient method has been developed for the conversion of black-oil simulator results into equivalent compositional results. This information is of value when aggregating production from many different sources and as input to process simulators. The **BOz** method is based on full utilization of compositional information usually thrown away while generation of PVT property tables for a black-oil simulator.

The method has been adapted for use in a generic split factor formulation. This allows easy implementation in computer programs e.g. in Streamz. Its implementation in Streamz has been used to evaluate the accuracy on numerous synthetic and actual field cases. This has confirmed its accuracy in all cases tested so far including depletion and gas injection. The bottom line in use of this method is the accuracy of performance prediction (surface rates and pressure profiles) by the black-oil model. If the black-oil model can be used with confidence for a particular case, BOz conversions can be used to reproduce accurate compositional predictions. Worst case scenarios from the benchmark paper by Fevang et. al.<sup>1</sup> have been used to validate this method.

Various issues related to the implementation of this method are discussed and explained. The adaptations for multiple PVT regions and gas injection cases have been explained. The potential errors expected from an incorrect use (e.g. field level) have been illustrated. The limits of this method are investigated and guidelines are provided to avoid inaccuracies and non-physical results.

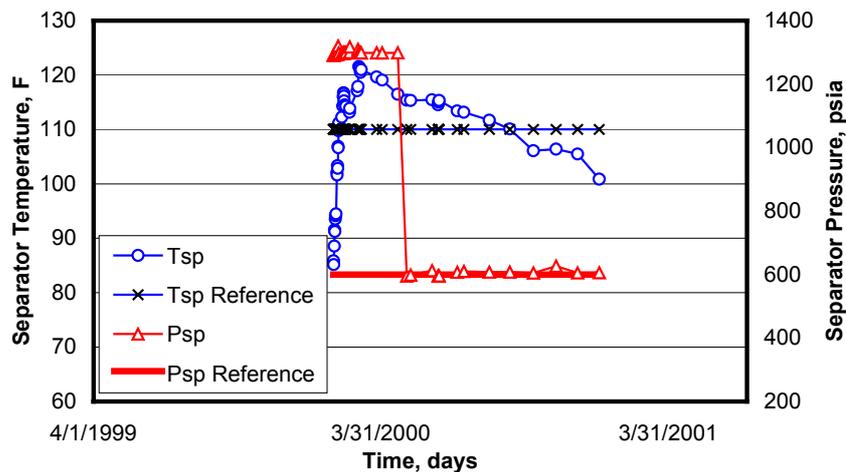
## 2.4 Thermodynamic Conversions

Any other method that allows the calculation of amounts of destination stream component from source stream components is consistent with the generic split-factor formulation of section 2.1. Thermodynamic conversions use some thermodynamic method (e.g. an equation of state calculation) to obtain the split factors for the conversion. The generation of split factors may be done externally or may be done “on-the-fly”.

This section discusses some of these conversions. While some of this is already implemented in Streamz, a couple like the Leibovici et al and the SDInverse Lumping method also fit into Streamz’s generic stream conversion setup. This section describes these methods briefly and discusses how they may be incorporated into Streamz.

### 2.4.1 Split Factor based Well Test Conversions

This section develops the split factors for conversion of *reported* well test rates to rates at a “common” set of separator conditions.



**Fig. 2.65 – During a well test separator temperature and/or pressure varies making it difficult to interpret test results. A set of reference conditions must be chosen and the rates corrected to them.**

During Well Testing, separator rates are dependent on  $p_{sp}$ ,  $T_{sp}$ . Separator conditions  $p_{sp}$ ,  $T_{sp}$  vary during testing (Fig. 2.65)

- Continuously

- Abruptly

Reported rates may be difficult to analyze because of lack of consistency (e.g. GOR variation). Conversion of  $q_{osp}$  to STO  $q_o$  depends on shrinkage factor (SF).  $SF_{test}$  is usually constant and unreliable. One needs to use  $q_{osp}$  actually measured, but  $q_o$  STO is only reported (sometimes). Rates reported on a common set of separator conditions are needed to evaluate consistently the variation in produced wellstream composition (total GOR) for

- Sample validity.
- Identify coning.
- Identify if  $p_{wf} < p_{sat}$ .

The multitude of different rates (streams) being discussed in this section necessitates their accurate definition. Table 2.10 lists and defines these streams.

**Table 2.10 – Definition of streams discussed in this section**

Term	Definition
$q_{gsp}$	Reported Well Test separator gas rate at SC.
$q_{o\ test}$	Reported Well Test oil rate at SC.
$q_{osp}$	Reported Well Test separator oil rate at separator conditions.
$q_o$	Oil rate at standard conditions (same as STO).
$q_{g+}$	Additional gas produced by flashing the separator oil to standard conditions.
$q_{o\ ref}$	ST Oil rate if wellstream is taken through hypothetical single stage separator only instead of the Test Separator system.
$q_{g\ ref}$	Gas rate if wellstream is taken through hypothetical single stage separator only instead of the Test Separator system.
$R_{s+}$	Solution gas oil ratio of the separator oil resulting in $q_{g+}$ ( $\equiv q_{g+} / q_o$ ).
$B_{osp}$	Volume factor at separator conditions ( $\equiv q_{osp} / q_o$ ).
$B_{osp\ test}$	Test volume factor reported with test oil rate. Inverse of test shrinkage factor ( $SF_{test}$ ).
$SF(b_{osp})$	Shrinkage factor. Inverse of $B_{osp}$ .
$SF_{test}$	Test shrinkage factor. Usually calculated once and used to convert actual separator rates ( $q_{osp}$ ) to reported well test oil rate ( $q_{o\ test}$ ). Inverse of $B_{osp\ test}$ .
$p_{sp}$	Separator pressure.
$T_{sp}$	Separator temperature.

Fig. 2.66 depicts a typical test separator system, Fig. 2.67 a hypothetical single-stage separator, and Fig. 2.68 a common conditions test separator system. The conversion of measured well test data (Test Stream) to a corresponding stream at a common set of conditions is performed in a step-wise manner with a number of intermediate streams and conversions. Table 2.11 names the streams and constituent components (as defined in Table 2.10):

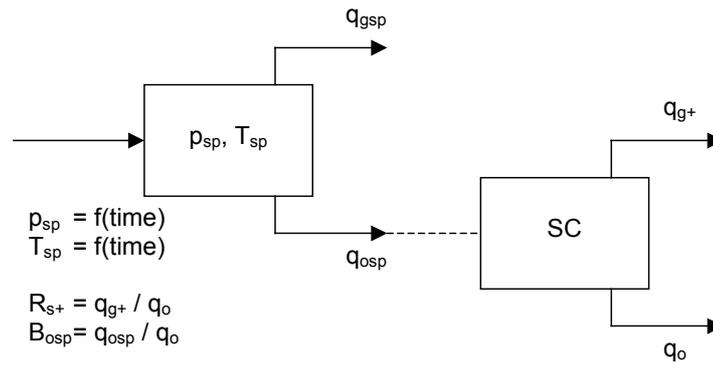


Fig. 2.66 – Test separator schematic and associated definitions.

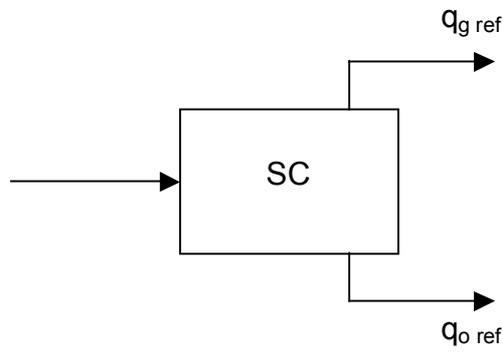


Fig. 2.67 – Hypothetical Single-Stage separator schematic and associated definitions.

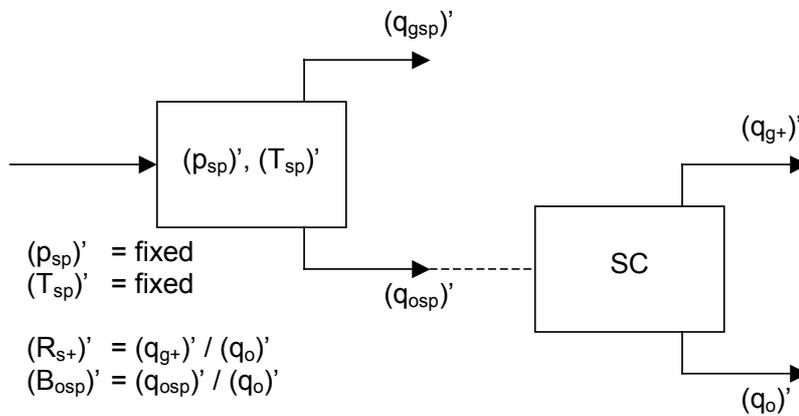


Fig. 2.68 – Common conditions separator schematic and associated definitions.

**Table 2.11 – Streams and constituent components**

StreamName	Components		
Test	$q_{o \text{ test}}$	$q_{gsp}$	
(A)	$q_{osp}$	$q_{gsp}$	
(B)	$q_o$	$q_{gsp}$	$q_{g+}$
(C)	$q_o$	$q_g$	
(D)	$q_{o \text{ ref}}$	$q_{g \text{ ref}}$	
(C)'	$(q_o)'$	$(q_g)'$	
(B)'	$(q_o)'$	$(q_{gsp})'$	$(q_{g+})'$
(A)'	$(q_{osp})'$	$(q_{gsp})'$	

The conversions are performed in the following sequence:

Test  $\rightarrow$  (A)  $\rightarrow$  (B)  $\rightarrow$  (C)  $\rightarrow$  (D)  $\rightarrow$  (C)'  $\rightarrow$  (B)'  $\rightarrow$  (A)'

The following equations are applicable for the conversions. Split factors are basically implementations of these equations:

**General equations**

$$q_g = q_{gsp} + q_{g+} \dots\dots\dots(2.80)$$

$$B_{osp} = \frac{q_{osp}}{q_o} \dots\dots\dots(2.81)$$

$$R_{s+} = \frac{q_{g+}}{q_o} \dots\dots\dots(2.82)$$

**Split Factor Conversion Equations**

Each component of the output stream will be written in terms of components of input stream.

Stream “Test” to stream (A)

$$q_{osp} = q_{o \text{ test}} \left( \frac{1}{SF_{\text{test}}} \right) + 0 \cdot q_{gsp} = B_{o \text{ test}} q_{o \text{ test}} + 0 \cdot q_{gsp} \dots\dots\dots(2.83)$$

$$q_{gsp} = 0 \cdot q_{o \text{ test}} + 1 \cdot q_{gsp} \dots\dots\dots(2.84)$$

Stream (A) to Stream (B)

$$q_o = \left( \frac{1}{B_{osp}} \right) q_{osp} + 0 \cdot q_{gsp} \dots\dots\dots (2.85)$$

$$q_{gsp} = 0 \cdot q_{osp} + 1 \cdot q_{gsp} \dots\dots\dots (2.86)$$

$$q_{g+} = \left( \frac{R_{s+}}{B_{osp}} \right) q_{osp} + 0 \cdot q_{gsp} \dots\dots\dots (2.87)$$

Stream (B) to Stream (C)

$$q_o = 1 \cdot q_o + 0 \cdot q_{gsp} + 0 \cdot q_{g+} \dots\dots\dots (2.88)$$

$$q_g = 0 \cdot q_{osp} + 1 \cdot q_{gsp} + 1 \cdot q_{g+} \dots\dots\dots (2.89)$$

Stream (C) to Stream (D)

$$q_{oref} = \frac{1}{f_o} \cdot q_o + 0 \cdot q_g \dots\dots\dots (2.90)$$

$$q_{gref} = 0 \cdot q_o + \frac{1}{f_g} \cdot q_g \dots\dots\dots (2.91)$$

The factors  $f_o$  and  $f_g$  are ratios of “oil (or gas) volume processed through any Test Separator system” to “oil (or gas) volume processed through a hypothetical Single Stage Separator system”. Their development is detailed in Appendix A.

Stream (D) to Stream (C)'

$$(q_o)' = (f_o)' \cdot q_{oref} + 0 \cdot q_{gref} \dots\dots\dots (2.92)$$

$$(q_g)' = 0 \cdot q_{oref} + (f_g)' \cdot q_{gref} \dots\dots\dots (2.93)$$

The factors  $(f_o)'$  and  $(f_g)'$  are the reciprocals of  $f_o$  and  $f_g$ . The development is detailed in Appendix A.

Stream (C)' to Stream (B)'

$$(q_o)' = 1 \cdot (q_o)' + 0 \cdot (q_g)' \dots\dots\dots(2.94)$$

$$(q_{gsp})' = (-R_{s+})' \cdot (q_o)' + 1 \cdot (q_g)' \dots\dots\dots(2.95)$$

$$(q_{g+})' = (R_{s+})' \cdot (q_o)' + 0 \cdot (q_g)' \dots\dots\dots(2.96)$$

Stream (B)' to Stream (A)'

$$(q_{osp})' = (B_{osp})' \cdot (q_o)' + 0 \cdot (q_{gsp})' + \left( \frac{(B_{osp})'}{(R_{s+})'} \right) \cdot (q_{g+})' \dots\dots\dots(2.97)$$

$$(q_{gsp})' = 0 \cdot (q_{go})' + 1 \cdot (q_{gsp})' + 0 \cdot (q_{g+})' \dots\dots\dots(2.98)$$

The multipliers (e.g.  $B_{osp}$  in Eq. (2.97)) to individual stream components (e.g.  $(q_o)'$  in Eq. (2.97)) are exactly in the form of Streamz split factors. They can be calculated by performing an EOS calculation. They are calculated for a range of separator temperature and pressures and interpolated as required.

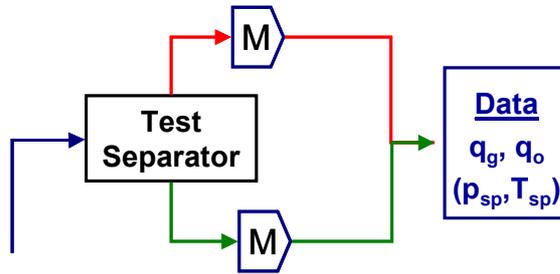
#### 2.4.2 Streamz Process command based Well Test Conversions

This section describes an alternate method for conversion of *reported* well test rates to rates at a “common” set of separator conditions. The background discussion in the previous section 2.4.1 still applies. The multitude of different rates (streams) being discussed in this section necessitates their accurate definition. The definitions in Table 2.10 will be used.

The conversion of measured well test data (Test Stream) to a corresponding stream at a common set of conditions is performed in a step-wise manner with a number of intermediate streams and conversions. The initial stream consists of the measured test rates  $q_g$  and  $q_o$ , and the separator conditions  $p_{sp}$  and  $T_{sp}$  as depicted in Fig. 2.69.

Usually the incoming feed is sampled to give an initial estimate of the feed composition. Alternatively, the gas and oil streams are sampled and recombined in the ratio of the GOR. Either gives a good estimate of the feed composition. This feed is flashed through a Simulated Separator Test as shown

in Fig. 2.70. This gives calculated volumes of gas and oil  $q_g^c$  and  $q_o^c$ . We want to honor the measured rates and modify the feed estimate to one that will produce the correct measured rates. This correction is done based on Eq. (2.99).

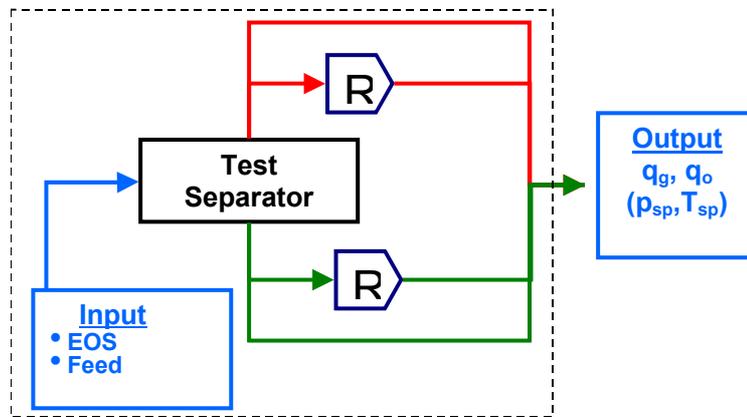


**Fig. 2.69** – Representation of the Test separator system and data recorded during one.

$$z_i = n_o x_i + n_g y_i \dots\dots\dots (2.99)$$

$$z_i = \frac{n_o^m}{n_o^c} n_o^c x_i + \frac{n_g^m}{n_g^c} n_g^c y_i \dots\dots\dots (2.100)$$

$$z_i \approx \frac{q_o^m}{q_o^c} n_o^c x_i + \frac{q_g^m}{q_g^c} n_g^c y_i \dots\dots\dots (2.101)$$



**Fig. 2.70** – Representation of the Simulated Test separator system and its input and output.

To obtain a corrected feed estimate we need to scale the oil moles by the ratio  $q_o^m/q_o^c$  and the gas moles by the ratio  $q_g^m/q_g^c$ . This recombination of

corrected oil and gas moles gives the corrected feed estimate. This can then be taken through a Test Separator system at the common  $p_{sp}$  and  $T_{sp}$ . The resulting oil volume and gas volumes give the corrected rates as depicted in Fig. 2.71.

The author of Streamz, A. A. Zick, has shown<sup>14</sup> the implementation of Well Test rates correction within Streamz. It makes use of the very generic and flexible nature of the “Process” command in Streamz. A description of the Process module and its use to solve this problem is discussed in Chapter 3.

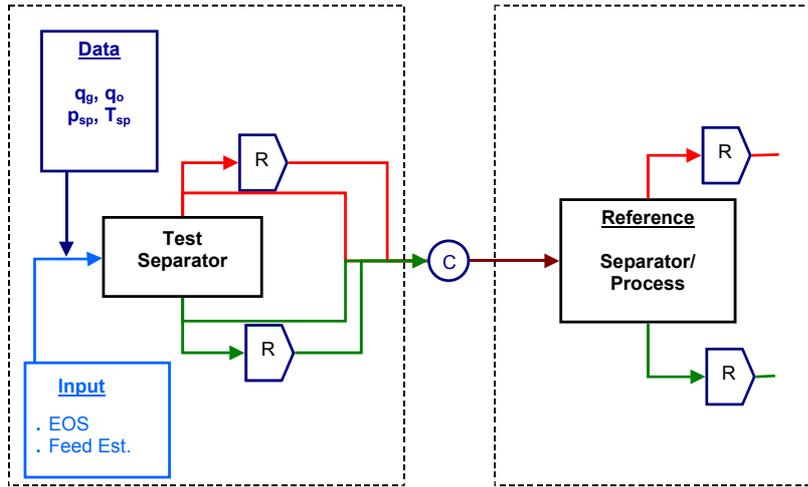


Fig. 2.71 – Schematic of feed estimate correction and its use to obtain test rates at common conditions (Reference).

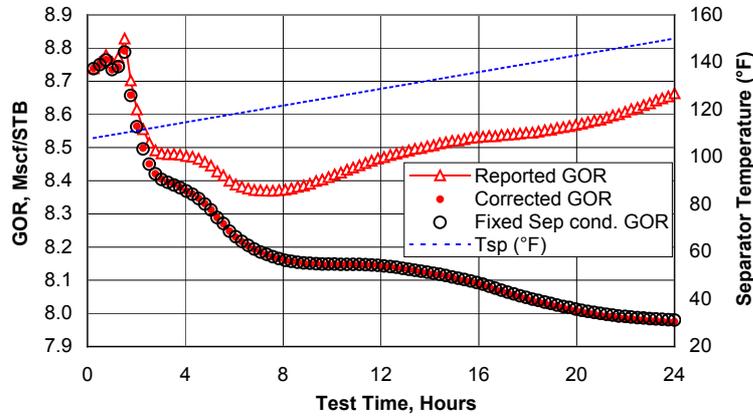


Fig. 2.72 – GOR Correction for Varying Separator Conditions for rich GC producing below DP with varying separator temperature. Excellent match between simulator re-run and PSM/Streamz processed GOR correction at common conditions (Reference).

A variable separator condition well test was simulated to check the validity of this stream “conversion”. The reservoir simulator Sensor<sup>15</sup> allows the changing of surface separator conditions. The resulting oil and gas rates correctly mimic Well Test rates under this varying condition. A case with a constantly varying temperature (fixed pressure) is shown in Fig. 2.72. The open triangle symbols with thin line shows that the GOR has a general increasing trend. When corrected with by applying Streamz process command, the solid circle symbols show a general downward trend instead.

To cross-check the GOR corrections from Streamz, the Sensor was re-run using the same set of common separator conditions. The match is excellent as depicted by large open circles in Fig. 2.72.

### 2.4.3 Delumping

Delumping is a general term frequently used in the conversion of pseudo-component quantities into pure component quantities. As described elsewhere<sup>1,2</sup>, pseudo-components are used in compositional reservoir simulators for computational efficiency. While many authors<sup>1,2</sup> have discussed pseudoization, all procedures required the grouping pure components into pseudo-components.

Coats<sup>2</sup> distinguishes between the terms *grouping* and *lumping*. Grouping occurs where each pseudo-component includes some of each of the  $n$  components in the original mixture. In lumping components, each pseudo-component consists of a subset of the original components and none of the members of this subset are present in any of the other pseudo-components. For example, Mixture  $z$  of  $n = 8$  components,  $\text{CO}_2$ ,  $C_1$ ,  $C_2$ ,  $C_3$ ,  $C_4$ ,  $C_5$ ,  $C_6$ , and  $C_{7+}$ , might be pseudoized to  $m = 5$  pseudo-components,  $x^1 = C_1$ ,  $\{x^2 = \text{CO}_2, C_2\}$ ,  $\{x^3 = C_3, C_4\}$ ,  $\{x^4 = C_5, C_6\}$ , and  $x^5 = C_{7+}$ .

While lumping components together is trivial (simple addition), de-lumping is not so. The simplest de-lumping is based on the trivial lumping, but this is only valid for the initial feed because it requires the availability of a sample of the detailed components. In the absence of any other information, this de-lumping could be used for the lighter components for water injection.

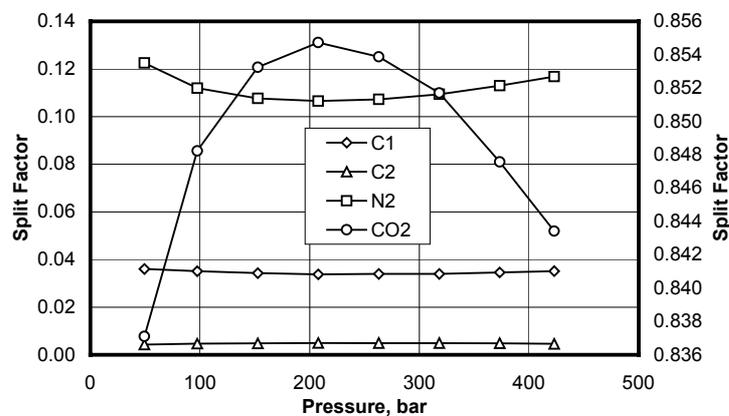
#### De-lumping based on Depletion-type Experiment

As a first approximation to incorporate a variation in the de-lumping factors (split factors), a depletion-type (CVD, DLE, CCE) experiment is performed in the PVT program. This generates a set of split factors that vary with pressure. A CVD experiment approximates the depletion of a gas condensate quite well.

**Table 2.12 – CVD results obtained during a PVT Study on a North Sea Rich Gas Condensate reservoir sample.**

Component	423 psia	373.3 psia	318.2 psia	263 psia	207.8 psia	152.7 psia	97.5 psia	49.3 psia
CO2	2.74	2.85	2.9	2.97	3	3.07	3.14	3.17
N2	0.36	0.4	0.42	0.43	0.44	0.43	0.42	0.38
C1	65.77	69.84	72.71	74.68	75.9	76.4	75.9	73.64
C2	9.11	9.31	9.34	9.38	9.46	9.64	10.02	10.78
C3	4.91	4.84	4.68	4.58	4.62	4.72	5.03	5.69
IC4	0.69	0.66	0.63	0.6	0.6	0.61	0.66	0.75
C4	1.9	1.79	1.68	1.6	1.6	1.63	1.71	2.03
IC5	0.55	0.51	0.46	0.43	0.41	0.42	0.46	0.51
C5	0.79	0.72	0.64	0.6	0.58	0.58	0.62	0.69
C6	0.94	0.82	0.71	0.64	0.6	0.57	0.58	0.62
C7	1.6	1.47	1.29	1.11	0.93	0.79	0.66	0.77
C8	1.77	1.34	1.13	0.95	0.78	0.6	0.46	0.57
C9	1.04	0.7	0.55	0.39	0.3	0.21	0.15	0.18
C10	0.79	0.58	0.38	0.24	0.15	0.08	0.05	0.07
C11+	7.04	4.17	2.48	1.4	0.63	0.25	0.14	0.15

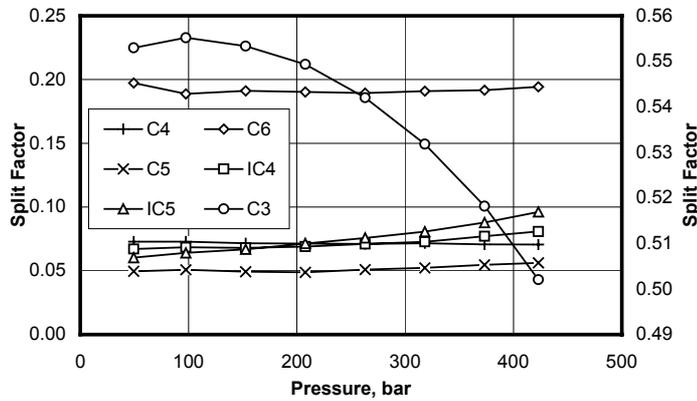
Table 2.12 contains results from a CVD experiment conducted during a PVT Study on a representative sample from a rich gas condensate field in the North Sea. The EOS model developed for the fluids in this field contained 6 pseudo-components with 4 plus fractions. The PVT tables for black-oil simulation were generated from this 6-component model. BOZ conversion was applied to simulation results to obtain 6-component molar rates. It was then required to convert this to a 17-component process characterization. While the Gamma Model was used on the plus fractions a method for de-lumping the 2 minus pseudo-components into 10 pure components was needed.

**Fig. 2.73 – Variation of CVD based split factors as a function of pressure for the light components.**

Since a CVD experiments favorably approximates a Gas Condensate field producing under depletion, it was decided to use the experimental CVD data.

The split factors, correlated to pressure, were calculated by the trivial lumping of components:  $X1 = CO_2 + N_2 + C_1 + C_2$  and  $X2 = C_3 + IC_4 + C_4 + IC_5 + C_5 + C_6$ .

Fig. 2.73 and Fig. 2.74 show the split factors for, respectively, the light and intermediate components as a function of pressure. These were used in the “HTHP Rich Gas Condensate Reservoir” example in section 2.3.8.



**Fig. 2.74 – Variation of CVD based split factors as a function of pressure for the intermediate components.**

#### De-lumping based on small-scale model results

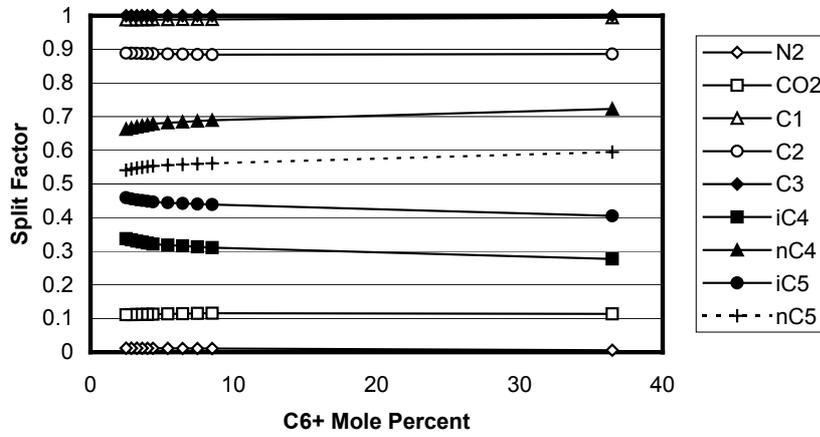
While PVT experiment based split factors capture only a limited dependence, a reduced (e.g. a pilot) model captures more factors affecting the performance of a real field. A compositional dependence can be captured by correlating the variation of split factors to some compositional property (e.g.  $C_{6+}$  amount) of the streams.

Results from a sector model of a North Sea field run using a 16-component model were available. The results were used to calculate the split factors for an 8-component EOS model that was being used for the full field models. The split factors were correlated with  $C_{6+}$  mole percent.

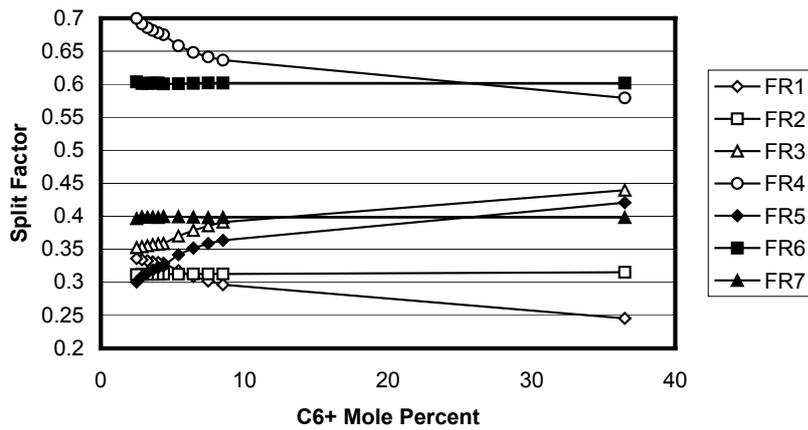
For the Sector Model the  $C_{6+}$  amount varied from 8.5070 to 2.4824 mole percent. For the full field case, the initial  $C_{6+}$  amount was 36.48 mole percent. Split factors for this amount of  $C_{6+}$  were generated using trivial lumping. No data were available between  $C_{6+}$  of 8.5070 to 36.48. A plot of split factors versus  $C_{6+}$  provides insight to whether linear interpolation could be used within this range.

Fig. 2.75 is a plot of split factors of  $C_5$  versus the  $C_{6+}$  amount. The trend for each component (considering only the last 4 points from the sector model and

the single point in the full-field model) is essentially a straight line. Hence linear interpolation of split factors between the  $C_{6+}$  values of 8.5070 to 36.48 mole percent would give the best estimate. So we use the combination of split factors from the Sector model and the initial feed of the full feed model.



**Fig. 2.75 – Variation of Sector Model based split factors as a function of  $C_{6+}$  mole percent for the  $C_5$ - components.**



**Fig. 2.76 – Variation of Sector Model based split factors as a function of pressure for the  $C_{6+}$  components.**

Fig. 2.76 is the split factor vs.  $C_{6+}$  amount plot for the 7  $C_{6+}$  fractions. FR1, FR3, FR4 and FR5 all have non-linear trend (considering the last 4 points of the Sector model and the single point of the full-field model). Fitting of various trends in MS Excel revealed that for F4 and F5 the deviation from

linear for intervening points was minimal. Hence FR3 was only chosen for fitting to a power series. Only the last 4 points on the sector model and the single point in the full-field model were chosen for this fit giving the function  $y = 0.3937x^{-0.1316}$ . Values of split factors for FR3 were calculated using this function for intervening values (10, 15, 20, 25, and 30). Since FR1, FR2 and FR3 make up the single pseudo-component F123, and FR2 had a linear trend, values of FR2 were obtained using linear trend and the values for FR1 were calculated as  $FR2 = 1 - FR1 - FR3$ .

This discussion illustrates simple engineering required to come up with the “best” solution for a particular situation. Once developed, a set of split factors could be used for all full-field simulation de-lumping using Streamz.

It is, however, clear that such a correlation to a compositional property for de-lumping is not universally applicable. It is usually applicable to the particular field or portion of the field.

### LSK Delumping

The problem of conversion from a pseudoized reservoir compositional representation to a detailed (process) compositional characterization has received much attention. Recently Leibovici et. al.<sup>3</sup> have described a method for this conversion, which they call delumping. It is based on the observation that, for a system with no binary interaction coefficients (BICs), the equilibrium k-value ( $K$ ) is related to the EOS parameters ( $a_i$  and  $b_i$ ) through the linear relationship

$$\ln(K_i) = C_0 + C_1\sqrt{a_i} + C_2b_i \dots\dots\dots(2.102)$$

Here the constants  $C_j$  depend only on pressure and temperature. Thus, provided that the flash made with the lumped fluid provides a good approximation to the phase parameter values, the constants will be approximately the same for both the detailed and the lumped fluid models.

For a single flash delumping, they first perform the flash with the lumped fluid model to obtain the equilibrium k-value for each component. These are then used to determine the constants  $C_j$  by regression using equation (2.102). The same equation (2.102), now using the EOS parameters ( $a_i$  and  $b_i$ ) for the detailed fluid model, then gives the equilibrium k-value  $K_i$  for each detailed component. The vapor mole fraction from the pseudo fluid model flash can be used, or recomputed using Rachford-Rice, to calculate the mole fractions of the detailed components.

Equation (2.102) is exact for zero BIPs and regression is needed for non-zero BIPs. In either case at least 3 lumped components must be present. The

quality of this procedure is also dependent on the accuracy of the lumped EOS, i.e. how well it matches predictions of the detailed EOS.

Progressing from the single flash delump to a full-fledged Reservoir Simulation delump requires delumping every flash performed by the simulator. In other words the flash in each grid cell for each time-step is needed, as the feed to the grid-cells in the well come from those in adjacent cells, and so on. They assert that such a procedure is computationally more efficient than running the simulation with the detailed EOS. To perform such de-lumping as a post-processor requires huge amount of space to store the required data.

This procedure has also been incorporated as an option into Eclipse 300.

### SD Inverse Lumping

A.G. Schlijper and J.K. Drohm<sup>4</sup> discuss methods of translating group (pseudocomponent) compositions into detailed component compositions, a procedure they refer to as “inverse lumping”. They propose its use in calculation of phase properties in the group model and for inverse lumping the well streams as predicted by the simulator so that the results can be used in a fully compositional separator-system (process) simulator.

They discuss two separate methods. In the first part an inverse-lumping procedure that can be applied if the grouped thermodynamic model and the original component model use the same EOS. They also discuss applications if the simplified model is a tabular or diagrammatical (usually with ternary diagrams) representation of pre-calculated phase equilibria. Only the first application is discussed in this section.

The basic approach is to use an approximation,  $K_i^\circ$ , to the expression for the actual equilibrium k-value,  $K_i$  as given in Eq. (2.103):

$$\begin{aligned} \log(K_i) = & \left[ F_{1,L}(p, T, a_L^\circ, b_L^\circ) - F_{1,V}(p, T, a_V^\circ, b_V^\circ) \right] \\ & + \left[ F_{2,L}(p, T, a_L^\circ, b_L^\circ) \sum_j \sum_t S_{jt} x_t^\circ a_{ij} \right. \\ & \left. - F_{2,V}(p, T, a_V^\circ, b_V^\circ) \sum_j \sum_t S_{jt} y_t^\circ a_{ij} \right] \dots\dots\dots (2.103) \\ & + \left[ F_{3,L}(p, T, a_L^\circ, b_L^\circ) b_i - F_{3,V}(p, T, a_V^\circ, b_V^\circ) b_i \right] \end{aligned}$$

This removes the equilibrium K-values dependence on detail component mole fractions and is completely determined by the group mole fractions,  $x_s^\circ$  and  $y_s^\circ$ , split parameters  $S_{is}$  and EOS parameters  $a_{ij}$  and  $b_i$ . Thus having solved a

vapor/liquid equilibrium problem in the group model, they use the resulting values for  $X^o$ ,  $Y^o$  (liquid and vapor fractions for the group model),  $x_s^o$  and  $y_s^o$  (group mole fractions) to calculate the approximations,  $K_i^o$ , to the component  $K$  factors at equilibrium. Determination of component phase mole fractions,  $x_i$  and  $y_i$  from the equation,

$$y_i = K_i^o x_i \dots\dots\dots(2.104)$$

and the material balance equations,

$$x_i X + y_i Y = z_i = \sum_s S_{is} z_s^o \dots\dots\dots(2.105)$$

is a standard fixed  $K$ -value problem and is effected by solving iteratively one equation in one unknown.

Implicit in their procedure for inverse lumping is the need for what they term the split parameters,  $S_{is}$ . This is exactly analogous to the Split Factor defined and used throughout this chapter. Quoting from their paper "These parameters carry the information on how to split the group mole fractions,  $z_s^o$ , of the fluid feed mixture into the component feed mole fractions,  $z_i$ ...To predict the  $S_{is}$ , therefore, it would be necessary to know in advance the composition path that the reservoir fluid mixture would take in the course of the recovery process. Because this knowledge is obviously not available, the  $S_{is}$  must be estimated on the basis of the expected range of feed compositions expected to be encountered."

Since the  $S_{is}$  need to be estimated from before and needs to be used, it is not at all clear why the extra, complicated, steps in determining the phase compositions are required at all. It should be straightforward to use the group mole fractions,  $z_s^o$ , to determine the detailed mole fractions,  $z_i$ . This is the approach used in PSM.

It might be argued that the relationship between the group and detailed components is more easily and accurately correlated for the phase compositions rather than the wellstream composition. This relationship is used in Eq. (2.103). But Eq. (2.105) does use the relationship for wellstream composition anyway.

The comment "This same expected range is used in preparing the group-model description" gives an interesting idea for generation of the split factors. Since it is always recommended to use samples covering the expected composition-pressure-temperature space during the development of the detailed and the group (pseudoized) model, it might be possible to

systematically generate the set of split factors during this pseudoization process. All practical dependencies can be incorporated into one or more correlating variables. This would be an ideal solution in the absence of any other means of generating split factor dependence (e.g. sector models or Gamma fit).

#### 2.4.4 Conclusions

This section has discussed the varied methods of generating split factors where some EOS based calculation is used to generate required data. Well test conversions based on Streamz **process** command form a universally applicable solution.

Various problem-specific de-lumping solutions are described and successfully applied on field problems. The Lebovici et. al.<sup>3</sup> and Schlijper<sup>4</sup> methods are attempts on universally applicable methods for the de-lumping type conversion of Petroleum streams.

Neither of these seems to represent a general solution to the de-lumping problem. Further research is warranted to find better solutions.

## Chapter 3

# Streamz Command Syntax & Examples

### 3.1 Introduction

The Streamz program was used exclusively at all stages during this research. It is an extremely sophisticated piece of software programmed in C++. For the full utilization of Streamz it was necessary to understand the internal working of the program. This chapter introduces the basic concepts that are needed to use Streamz.

Streamz has an extremely flexible structure and can be used for a variety of stream conversion problems. Using a small set of 31 primary commands it is possible to perform virtually any conversion. A short description of these primary commands is presented. Streamz has been used in all the example conversions discussed in Chapter 2. This Chapter will describe all the data sets used in those examples.

### 3.2 Streamz Concepts

#### 3.2.1 Characterization

Streamz is a program developed to facilitate conversions among petroleum fluid stream representations. A form of stream representation in Streamz is called a **characterization**. The user only has to a) name the characterization, and b) name the components comprising the characterization. An equation of state (EOS) characterization is acceptable but is simply a super-set where the user additionally supplies typical EOS critical properties such as critical temperature, critical pressure etc. These are only required if the stream happens to participate in a thermodynamic conversion.

This flexibility in defining of characterizations allows one to define non-typical non-petroleum characterizations like:

```
Characterization Fruits
Names
Mango
Guava
Lichee
```

The above extract of a Streamz data-set declares a **characterization** named “Fruits”. It then **names** the 3 components (sub-keyword **Names** and **Components** can be used inter-changeably). For the purpose of discussion we declare another non-typical stream characterization as:

```
Characterization TotalPrice
  Name
    Rupees
End
```

At the other extreme, a full-blown EOS characterization would include all the critical properties (e.g. Critical pressure, Critical temperature etc.) and binary interaction parameters that are essential for performing accurate EOS calculations.

### 3.2.2 Stream

A stream is always linked to a characterization. A stream, in its most basic form, is a collection of quantity data for each component in the characterization. The flexible nature of Streamz allows for the quantity data to be entered in units of amounts. Streamz recognizes special units of **moles**, **mass** and **volumes** for performing special (e.g EOS) calculations.

In addition to quantity data, it is also possible to associate attributes to a stream. These attributes are called **variables**. Typical usage of these variables would be to associate a name or a pressure with the stream. The utility of these variables are two-fold: a) They allow streams to be **converted differently** based on values of these variables, and b) They allow streams to **filtered** based on the values of these variables.

Normally a stream is a line of data in a stream file. Multiple streams may reside in the same file if they belong to the same characterization. This is typical of reservoir simulator streams where tens of thousands of streams reside in the same stream file.

Building upon the “Fruits” characterization in section 3.2.1 a stream could be written thus:

```
Streamz 1
Characterization "Fruits"
Data
Amounts
5      10      15
```

The keywords up to **Data** specify special information about all the stream data contained in the particular stream file. **Amounts** specifies that the actual stream data are in generic units. There are 5 amounts of the first component, 10 of the second, and 15 of the third.

### 3.2.3 Conversion

The flexibility of Streamz is extended to conversions. The generic conversion matrix based on split factors, introduced in section 2.1, allows any portion of an input component to partition into zero or more output components. This **split factor** need not be a fraction between 0 and 1, but is any number which would multiply the **amount** of that particular component to result in a portion of the output component as explained in section 2.1.

We can define a **conversion** from the Fruits characterization to TotalPrice as under:

```
Restore TotalPrice

Convert Fruits
  Split Mango Rupees 5
  Split Guava Rupees 2
  Split Lichee Rupees 4
End
```

Here we define the method to **Convert** Fruits to TotalPrice. Thus any stream of fruits (i.e. stream of numbers with the characterization “Fruits”) would result in a weighted sum of Rupees. For our example, a stream of 5 mangoes, 10 guavas and 15 lichees would result in 105 Rupees.

## 3.3 Streamz Commands

This section will present a categorized listing of the primary keywords understood by Streamz. They provide the primary user interface to perform the extensive and varied conversion, management and processing operations on petroleum fluid (rather any conceivable) streams. Detailed descriptions of these commands and all options with examples are available in the Streamz<sup>19</sup> documentation.

### 3.3.1 General Program and I/O Control Commands

#### Cd

The **Cd** keyword allows the user to change the current directory for inclusion of files or to open files from certain directories, without the need to specify the path along with the name of file. This keyword provides the functionality of the change directory command of many operating systems (OS). The argument to this command is a **path\_spec** that may include symbols typically used in “change directory” commands of supported OSes (Win9x/NT, DOS, Unix, Mac OS) like “.” for current directory, “..” for parent directory, “/” (or “\” or “:” depending on the OS) for directory name separators. If the **path\_spec** is to include embedded spaces, it should be enclosed within

quotes. Care should be taken while using these special symbols, as they may trigger errors if used on the wrong OS.

### **Define**

The **Define** command is used to associate a token (contiguous string) with a replacement string. This command allows a run-time replacement of any occurrence of the token, surrounded by question marks (?), with its associated string. All replacements will occur before any other parsing of an affected input line. The definition will persist throughout the rest of the driver file in which it is issued, carrying over into included files as well. It will not carry back into a parent driver file, however.

Used by itself, the **Define** command allows use of generic Streamz driver files on multiple cases, just by changing the replacement string for the token at one place.

In conjunction with the **Include** command, the **Define** command offers a very powerful utility for execution of the same set of generic instructions on the same token after it gets redefined (see section 3.4.3 for an example).

### **Echo**

This keyword instructs the program to write out *all* lines read from input (driver) file to the Standard Output (log). Normally Streamz only writes back certain information important to the run, some information about its execution, warnings and errors. By use of this keyword with the **On** option (or without any options which means the same as the **On** option) the user forces the program to write out all lines read from the input (driver) file. The written lines also contain information about the line number *in the input file*.

Use of the keyword with the **Off** option suspends the writing back of the read line for the rest of the execution of the program, unless instructed once again by use of the **On** option. The program initially starts with the **Echo Off** command in effect, without explicitly being specified.

### **End**

This keyword declares the end of the scope of the current primary keyword. When a primary keyword is in effect some sub-keywords and options may be recognized differently within its context. This keyword explicitly tells the program that no further sub-keywords to the current primary keyword are expected and the program should expect a primary keyword.

**Eof**

This keyword declares the “end of file” to the program. It is a trigger to the program that for all practical purposes, no information of importance exists after this keyword and the program can proceed with execution.

This is a useful keyword if the user has a large data-set, but wants only to execute an initial portion of it. If the **Eof** keyword is inserted at the particular point, the rest of data-set need not be deleted.

**Tabs**

**Tabs** keyword provides a means to specify the tab positions in the file. Normally spaces are used to separate individual words (records) in data files. If tab characters are used to separate individual records in the Input (driver) file, the argument specifies the position from which they will be read into the program. This reading of positions of each data is irrelevant to Streamz except in the tabular input format used for the input of EOS property (**Comp** command) and the BIP table (**Bips** command). These data are read in tabular format with the values getting associated with the headings depending on their “line up” (Refer to Appendix B for a discussion on the line up procedure).

**Title**

The purpose of the **Title** command is to print a boxed title to the Standard Output (log) file. The quoted string following the **Title** keyword is centered within a box made up of asterisk (\*) characters. The box expands to accommodate the full length of the string. This may be used to visually separate different tasks being run from the same input file.

**3.3.2 Fluid Characterization Commands****Characterization**

The **Characterization** keyword (**Char** for short; **Properties** is an alias) specifies a name for the characterization that is being defined. The actual description of the characterization is done using the **Components** (or its alias **Names**) and the **BIPS** keywords. The **Characterization** keyword makes the named characterization “current” and associates the property table (**Components** keyword) and the binary interaction parameters (**BIPS** keyword) with it.

**EOS**

This keyword declares the “global” Equation of State to which the next characterization applies (unless over-ridden by use of **EOS** sub-keyword within a **Characterization** command). Each characterization has to have an EOS associated. The allowed Equations of States are listed in Streamz User Manual<sup>19</sup>. The argument to this keyword specifies the EOS to be associated.

The default (modified Peng-Robinson 1979; PR) is associated if none is specified.

### **BIPS**

The BIPS keyword initiates the tabular input of the binary interaction parameters. The arguments following this keyword are predefined component names (in any order). Following this command line, the program expects a formatted table containing the values of binary interaction parameters, where the values “line up” (see Appendix B for details on line up procedure) with their heading.

### **Components**

The **Components** (or its alias **Names**) keyword triggers tabular input of the full EOS property table that makes up the current characterization. The arguments following this keyword are predefined properties for this characterization (in any order). This tabular input scheme is very flexible, allowing any of the component properties (including their names) to be input in any order. Each property is identified by a heading keyword. For example, MW indicates molecular weight. The only constraint of the tabular input scheme is that any entry in the table that belongs to a particular heading (i.e., property) should line up with the heading. This allows unknown entries to be left blank without misaligning the rest of the table. The specific rules for lining up the table are described in Appendix B.

### **Restore**

This keyword is used to make a previously defined characterization active. The argument to this keyword is the name of this previous characterization, which should be enclosed within quotes if it includes embedded spaces. It is frequently used prior to a **Convert** command to change the active characterization.

### **Lump**

This keyword defines a lumped fraction from among the components making up the current characterization and gives it a name. The first argument to this keyword is the name of the lumped fraction. Multiple arguments may then follow, each in the form of a doublet of component names (or previous lumps) and amounts. The idea is to define how much of each component makes up the **Lump**. Both elements of the doublet are optional. If the component name is omitted, it defaults to the one following the previous (with the first defaulting to the first named component in the active characterization). If the amount is omitted, it defaults to 1.

The utility of this keyword is to use a compositional property (e.g. total moles) of a single or combination of components as variables. This allows filtering,

converting and processing of streams based on current value of these lumps calculated on the fly.

### 3.3.3 Characterization Conversion Commands

#### **Convert**

The keyword defines completely the procedure to be used for conversions between a specified and the “currently active” characterization. It also specifies the units of input and output streams, and the quantity to be conserved in the conversion (if applicable). Currently two procedures are available: 1) Gamma, and 2) Split. The Gamma method uses the Gamma distribution procedure to first fit the input stream to the Gamma model and then to calculate the amounts in the output stream corresponding to this calculated model.

The second method is to supply a set of split factors that may be piece-wise linear functions of any number of variables. A combination of the two methods is also possible. This is usually the most typical procedure as the heptanes plus ( $C_{7+}$ ) components are best converted using the Gamma distribution and the hexanes minus ( $C_{6-}$ ) components are converted using split factors.

#### **Reduce**

This keyword is a special purpose **Convert** command to define the conversion from molar streams to volumetric streams. The argument identifies the name of the input characterization in which the molar (input) streams are expected. The active (output) characterization must be a single-component characterization. Additionally, the pressure and temperature has to be specified for this EOS-based conversion.

### 3.3.4 File Control Commands

#### **GammaFile**

The keyword associates a nickname with an actual file on disk and opens the file for output. This file receives Gamma distribution modeling results for the conversion where the same nickname is specified. The first argument is the nickname, the second identifies the purpose by means of an option (Open or Close), and the third identifies the actual file on disk.

#### **SplitFile**

The keyword associates a nickname with an actual file on disk and opens the file for output. This file receives all split factors calculated by the program during a conversion where the same nickname is specified. The syntax is similar to the **GammaFile** command.

### StreamFile

The keyword associates a nickname with an actual file on disk and opens the file for either input or output. The first argument command is the nickname, the second identifies the purpose by means of a keyword (Input, Output, Close), and the third identifies the actual file on disk.

Streams exist in these stream files and StreamFile command provides the only way of accessing them for conversion (or any other manipulation). Named streams can be created during execution of Streamz but they cease to exist once the program ends. StreamFile command provides the only way to save these named and converted streams to disk.

### 3.3.5 Variable and Domain definition Commands

#### Variable

This keyword defines a variable with a name and a type (string, numeric, or special).

The utility of this keyword is to declare and associate these declared **Variables** with streams. The association is accomplished generally in stream files by explicitly setting its value or by using the variable name as stream table headings. In either case some value is given to the particular variable and the variable, with the set value, becomes part of the stream. Streamz can then manipulate (convert, filter, process) these streams based on current values of these variables.

#### Domain

This keyword defines a special variable comprising of two existing normal **Variables** of the same type. It is used to specify an interval between these normal variables. If these two normal variables are associated with a stream, the resulting **Domain** is also associated with it automatically. The value of the **Domain** for a particular stream depends on the current values of the associated comprising the domain.

The utility of this keyword is the selection of accurate portions of streams when filtering them and is of considerable importance in the proper conversion from rates to cumulatives and back.

#### Set

The **Set** command is the only way to give values to declared variables and associate them with streams in stream files via a Streamz data set. The value includes the unit if the variable is one of the special types. Once set, the variables with its value becomes part of all the streams that follow.

### 3.3.6 Stream Manipulation and Output Commands

#### Combine

This keyword allows the aggregation of streams into new named streams. The **Combine** command creates a named stream by aggregating input streams based on filters (if any) and other options. A **Combine** command does not initiate output to a file, this is accomplished with a corresponding **Write** command.

#### Clear

This keyword is used to clear all Filters and named streams from memory. When used with optional sub-keywords it can also clear either one or the other.

When filters are defined, each new filter is stored in memory, even if the new one has the same name as a previous one. This is because the definition of the new filter itself may use the old one. In this way, each new use of the **Filter** command stores the filter and its logic in memory. This way it is available for use in any stream manipulation command. After a filter is used and no longer needed, the **Clear** command allows freeing of memory.

Similarly, named streams are stored in memory and can be cleared once they are no longer need for further execution of the program.

#### Copy

The **Copy** command initiates a read/convert/write operation of streams from all open input stream files to all open output stream files. Any conversion required during the operation, based on the characterizations associated with those files, is performed automatically. This is the recommended stream manipulation command when the original structure of the stream file is to be retained (along with all variables), and only a characterization change is needed.

#### Filter

This keyword defines and names a **Filter** (a set of criteria which must be true for it to be satisfied). The first argument to this keyword is the name of the filter being defined. Multiple (at least one) arguments may follow, each either a previously named filter (using previous **Filter** commands) or a filter construct, each separated by the sub-keywords **And**, **Or**, or **Not**. Previously defined **Variable** or **Domain** names are used and compared to their current values. Additionally, previously defined **Lumps** can be used.

Filters are the only interface for selecting only portions of streams (e.g. a particular well from a stream file containing streams from all the wells of a field) or for aggregating on a time basis (e.g. from every time step to yearly streams).

**Mix**

This keyword allows the creation of named streams by adding previously named streams and/or components. The first argument is the name of the stream being created. Multiple arguments can follow, each a triplet consisting of the 1) name of the stream or component being mixed, 2) the amount and 3) the unit. The destination stream will be assigned the current characterization. Any conversion required will be performed when the **Mix** command is executed.

This command allows the only way of creating streams without reading them in from existing stream files.

**Tag**

The **Tag** command associates defined **Variables** with a single named stream. The first argument to this command is the name of a stream (created previously in memory using **Combine**, **Total** or **Mix** commands). Multiple arguments can follow each of the form *var\_name var\_val*. This associates the variable of the supplied name with the stream and gives it the supplied value. The variable must have been previously defined using the **Variable** command. If the variable is one of the special types (temperature, pressure, time, or distance), the value must include the units.

The utility of this keyword is to assign variable and values to named streams after manipulation commands where the individual values of these variables in the component streams have been "unset" by the manipulation. Even otherwise it might be useful to associate variables and values to streams before they are written to stream file showing their origin (or some other relevant information).

**Tabulate**

The **Tabulate** command allows the user two specific functions.

- To sum out some of the variables associated with a stream and to create stream files containing only the variables asked for. All consecutive streams where the requested variables are constant are summed up (e.g. connection streams from a reservoir simulator may be summed up to wells).
- Create tabular stream files. All requested **Variables** are written out as table headings.

The **Tabulate** command also invokes the conversion if required. So it is recommended when some of the variables associated with the streams need

not be retained for further processing. The command takes as arguments previously defined variables, separated by **And**, which are to be retained.

### **Total**

The **Total** command allows the user to aggregate one or more previously created named streams. The first argument to this is the nickname of the named stream being created. The second argument is the sub-keyword **Add** followed by one or more stream names, each separated by the sub-keyword **And**.

The utility of this keyword is to easily combine streams previously created with different filters, into a single stream. This command does a simple adding of streams without any regard to their contents. It is the responsibility of the user to **Total** compatible streams.

### **Write**

This primary keyword is required only if named streams are created by the **Combine**, **Mix**, or **Total** commands, and they need to be written to files. This command can be requested to write multiple streams to multiple stream files.

## **3.3.7 Stream Processing Commands**

### **Process**

The purpose of the keyword is to allow separation of streams that pass through single or a series of interconnected separators. Coupled with **Reducers**, the **Process** command can be used to obtain volumetric rates from molar rates.

The keyword defines a complete process consisting of interconnected stages, each of which is a **Separator**, and also processes the input streams through it. All sub-keywords are optional but typically the **Stage** and **File** sub-keywords are always used. All needed conversions are performed automatically.

### **Separator**

The keyword names a Separator and its method of calculating the separation of incoming streams. It provides the functionality to Streamz of separating incoming feed streams into two product (liquid and vapor) streams. The possible separation methods are an **EOS** calculation, a set of **K-Val** tables, or a set of **Split** tables.

The user names the **Separator** and then uses named separators in **Stages** they define in the **Process** command. In absence of any **Process**, a **Separator** is non-functional. If no separation method is specified, **Split** is assumed.

Imaginative use of Streamz processing commands can provide Well Test GOR Correction modules, BOz re-processing modules, Surface Process Evaluation modules, and Condensate Allocation modules.

### 3.4 Streamz Commands used in Examples

Majority of the examples in section 2.3.8 is re-visited with the aim of explaining the Streamz commands that were used to generate the example. The first example will describe each and every command from the data-set while subsequent examples touch upon only the special/new commands for that example. For a full explanation of Streamz keywords the reader is referred to the proper program documentation<sup>18,19</sup>.

The majority of comments in the actual data sets are removed in the extract shown in this section to conserve space. The phrases “data set” and “driver file” are used interchangeably to refer to the text (ascii) file containing Streamz commands.

#### 3.4.1 North Sea Full-Field Case

Purpose of this example was to use Streamz on an actual field case to convert the black-oil results to compositional results. We then compare the results with those from an identical compositional simulation.

PSM linkers are used to generate the conversion table using the BOz method. PSM linkers also convert results in Eclipse format to stream files in Streamz format. We look at the Streamz data set used:

```
TITLE 'Streamz input file generated by PSM Toolkit'
INCLUDE 'bo.chr'
DEFINE inp_str_file = 'e100-bo.str'
STREAMFILE INP1: INPUT ?inp_str_file?
INCLUDE 'bo-9c-inj.cnv'
DEFINE out_str_file = 'e100-bo-9c.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?
END
INCLUDE 'tabulate.inc'
```

The **Title** command causes a boxed title to appear on the standard output (screen and/or file). The **Include** command causes the contents of the file specified as argument to be made part of the parent file. This file can have other Streamz commands and include other files. We look at the contents of this file before proceeding further (this is exactly what Streamz would do).

```
CHAR 'BO_Char'
NAME
SO
SG
```

The **Char** command is a short form of **Characterization**. This command informs Streamz that a characterization with the name “BO\_Char” is being declared. In this context Streamz understands many other sub-keywords (which might otherwise produce an error). One such keyword is what it encounters next, **Name**. It now enters its tabular-reading mode, where keywords are headings of a table and the corresponding data are to be found in the same column of the table. Each of the headings, including **Name**, forms the properties of the characterization being defined. When it next finds **SO** it associates it as the data corresponding to the property **Name** (i.e. name of the 1<sup>st</sup> component). It finds only one property (Name) and two components (SO and SG) for this characterization.

After at least one has been defined there will always be a “current” characterization in Streamz. It is the one defined last or the one made explicitly “current” using the **Restore** command.

Streamz then finds the end of the included file, closes it and proceeds further in the parent file.

```
DEFINE inp_str_file = 'e100-bo.str'
STREAMFILE INP1: INPUT ?inp_str_file?
```

The **Define** command defines a token (a contiguous string) and sets its value. The value acts as the replacement string whenever the defined token, enclosed within question marks (?) occur in the current or included files. This token is immediately used in the **StreamFile** command. This command specifies an operation on a stream file (a file containing streams; see section 3.2.2). The specific stream file is given a nickname **INP1** for use in the data set (the “:” is merely a word separator and is ignored). An **Input** operation is specified and the actual file on disk is specified using the previously defined token. **?inp\_str\_file?** translates to **e100-bo.str** and the file is opened by Streamz for input.

Whenever a stream file is opened Streamz links it to the “current” characterization and will read quantity data only for the number of components in that characterization. In stream file just opened for input should contain **Volume** data for the components **SO** and **SG**. In other words we open the stream file containing black-oil simulator data in Streamz format.

```
INCLUDE 'bo-9c-inj.cnv'
```

The user includes a file which, from the extension, appears to be a conversion file. Streamz, however, has no knowledge of file name nomenclature. We now look at this included file.

```

CHAR 'EOS9'
EOS PR
COMP MW      TC      PC      AF      VS      AMOD      BMOD
          K      BARA
C1  16.0924  190.115  45.9565  1.17E-02 -0.15913  1.00273  1.00243
CO2 44.01    304.206  73.8153  0.231    -8.20E-02 1        1
C2  30.07    305.422  48.8011  9.08E-02 -0.113    1        1
C3  44.097   369.822  42.4924  0.1454   -8.60E-02 1        1
C4  58.124   420.326  37.5454  0.187903 -7.18E-02 0.985158 1.00264
C5  72.151   465.861  33.7388  0.241262 -4.79E-02 1.00905  0.988291
C6F3 103.717  571.351  28.8451  0.315658 4.10E-02  0.966199 0.97905
F4   165.409 685.364  21.3935  0.479732 9.56E-02  1        1
F5F6 364.751 828.167  18.2002  0.710074 -0.42738  0.758841 0.927809

BIPS C1      CO2      C2      ...      C6F3      F4      F5F6
C1   0.00E+00 0.104245 7.19E-05 ... 2.77E-02 -1.04E-02 0.179524
CO2  0.104245 0.00E+00 0.13    ... 0.115    0.115    0.115
C2   7.19E-05 0.13    0.00E+00 ... 0.00E+00 0.00E+00 0.00E+00
C3   6.47E-04 0.125   0.00E+00 ... 0.00E+00 0.00E+00 0.00E+00
C4   6.83E-04 0.116424 0.00E+00 ... 0.00E+00 0.00E+00 0.00E+00
C5   7.61E-04 0.115   0.00E+00 ... 0.00E+00 0.00E+00 0.00E+00
C6F3 2.77E-02 0.115   0.00E+00 ... 0.00E+00 0.00E+00 0.00E+00
F4   -1.04E-02 0.115   0.00E+00 ... 0.00E+00 0.00E+00 0.00E+00
F5F6 0.179524 0.115   0.00E+00 ... 0.00E+00 0.00E+00 0.00E+00

```

This is a full-fledged EOS characterization including all critical properties (TC = Critical Temperature etc.) and the binary interaction parameters. A local EOS (PR) is linked to this characterization. When a BOz conversion is performed the compositional characterization to which BO rates are converted is always the same EOS characterization used to generate the BO PVT properties. This is the characterization being declared (with the name EOS9). This now becomes the “current” characterization for the data-set.

```

CONVERT 'BO_Char' from VOLUMES to MOLES

SET PVTNUM 1 PFLAG 2 PRES (BARA) 280.701
SPLIT SO C1
      -5.086E-01 1.903E-02 2.617E-01 2.801E-01 2.320E-01
      1.924E-01 1.162E+00 1.052E+00 1.458E+00
SPLIT SG C1
      3.768E-02 3.749E-04 2.598E-03 1.367E-03 7.048E-04
      3.370E-04 6.479E-04 -2.407E-05 -4.357E-04
.
.
.
SET PVTNUM 1 PFLAG 2 PRES (BARA) 40
SPLIT SO C1
      -4.530E-01 8.640E-03 1.923E-01 2.523E-01 2.255E-01
      1.969E-01 1.208E+00 1.073E+00 1.443E+00
SPLIT SG C1
      3.742E-02 3.805E-04 2.673E-03 1.438E-03 7.522E-04
      3.579E-04 4.703E-04 -1.812E-04 -2.824E-04
.
.
.
SET PVTNUM 1 PFLAG 1 PRES (BARA) 280.701

```

```

SPLIT      SO      C1
1.175E+00  -6.580E-02  -1.020E+00  4.374E-01  1.335E+00
1.006E+00  2.960E+00  9.848E-01  2.406E-01
SPLIT      SG      C1
3.708E-02  4.052E-04  3.057E-03  1.311E-03  3.098E-04
4.559E-05  4.221E-06  0.000E+00  0.000E+00
.
.
.
SET        PVTNUM 1  PFLAG 1  PRES      (BARA)  40
SPLIT      SO      C1
1.292E+00  -1.176E-01  -1.763E+00  8.995E-01  2.481E+00
1.789E+00  3.584E+00  1.492E-01  3.389E-03
SPLIT      SG      C1
3.708E-02  4.052E-04  3.057E-03  1.311E-03  3.098E-04
4.559E-05  4.221E-06  0.000E+00  0.000E+00
.
.
.
SET        PVTNUM 2  PFLAG 2  PRES      (BARA)  280.701
SPLIT      SO      C1
-3.807E-01  2.002E-02  2.671E-01  2.813E-01  2.316E-01
1.913E-01  1.151E+00  1.040E+00  1.440E+00
SPLIT      SG      C1
3.682E-02  3.686E-04  2.568E-03  1.365E-03  7.123E-04
3.485E-04  7.477E-04  7.948E-05  -2.836E-04
.
.
.
SET        PVTNUM 2  PFLAG 2  PRES      (BARA)  40
SPLIT      SO      C1
-4.514E-01  8.670E-03  1.927E-01  2.526E-01  2.258E-01
1.972E-01  1.210E+00  1.074E+00  1.445E+00
SPLIT      SG      C1
3.657E-02  3.731E-04  2.635E-03  1.432E-03  7.585E-04
3.699E-04  5.811E-04  -6.978E-05  -1.317E-04
.
.
.
SET        PVTNUM 2  PFLAG 1  PRES      (BARA)  280.701
SPLIT      SO      C1
7.181E+00  -2.658E-01  -4.017E+00  -2.326E+00  -1.942E-01
5.627E-01  3.632E+00  1.351E+00  3.301E-01
SPLIT      SG      C1
3.489E-02  4.417E-04  3.662E-03  2.031E-03  8.211E-04
2.536E-04  1.139E-04  0.000E+00  0.000E+00
.
.
.
SET        PVTNUM 2  PFLAG 1  PRES      (BARA)  40
SPLIT      SO      C1
1.789E+01  -7.397E-01  -1.102E+01  -6.288E+00  -4.573E-01
1.466E+00  6.310E+00  3.126E-01  7.098E-03
SPLIT      SG      C1
3.489E-02  4.417E-04  3.662E-03  2.031E-03  8.211E-04
2.536E-04  1.139E-04  0.000E+00  0.000E+00

```

Only an extract of the lengthy conversion table is reproduced. The **Convert** command defines the procedure for conversion from the specified characterization (BO\_Char in our example) to the “current” characterization (i.e. EOS9 in the example). The **from Volumes to Moles** specifies that the quantity data of the input streams should be in volumes and that of the output streams will be in moles.

The **Set** command specifies the conditions at which the following conversion procedure would be valid (and used). These conditions are the values of the variables associated with the particular stream being converted. The first **Set** command specifies the conditions of **PvtNum** (=1), **PFlag** (=2), and **Pres** (=280.701 bara).

**PvtNum** is a variable associated by one of the PSM Linkers to each stream based on which PVT region the stream corresponds to. The example reservoir model used 2 PVT regions and the split factors for each is different. Making the split factors a piece-wise linear function of the PvtNum integer allows each stream to use one set or the other.

**PFlag** is a variable linked to each stream allowing Streamz to use separate tables for a mixture of EG and IG and a mixture of EO and EG. This gas injection implementation is discussed in detail in section 2.3.3.

**Pres** is a variable linked to each stream. In most cases it is the block pressure of the grid-cell connection. Split factors for BOz conversion vary with pressure and by specifying various node pressures, we make the split factors a piece-wise linear function of pressure too. Since pressure is a special variable we need to specify the units.

Once the conditions (i.e. the values of the variables) have been set, the **Split** command specifies the actual split factor to be used at those conditions. The first argument (SO in example) is the input component for which the split factors are being specified. Following this, the 2<sup>nd</sup> argument (C1 in example) is the output component for which the split factors are specified. The next number (-5.086E-01 in example) is the split factor value. **Doublets** of output component names and split factor values are then expected for each output component. The example uses a short cut using only the split factor values. In such a case the component name defaults to the one following the previously used (CO2 following C1 in the example). Hence the 2<sup>nd</sup> number is the split factor for CO2. Numbers can continue on subsequent lines (as in the example). 9 numbers are used in the example hence each number is the split factor value for each component in the output characterization, in order.

A similar procedure is repeated for the gas split factors. Each split factor value fills the  $S_{ij}$  matrix and output molar rate for a particular stream would be

calculated using the same equation introduced in section 2.3.1 and reproduced here as Eq. (3.1).

$$m_i = \sum_{j=1}^2 S_{ij} q_j \dots\dots\dots (3.1)$$

It needs to be highlighted that, for now, Streamz merely stores this split factor matrix, the associated variable-value pairs, and the pair of characterizations in a conversion object. When it later encounters a command that initiates a conversion, Streamz will attempt to locate the valid converter, use the stored  $S_{ij}$  and fill in the  $q_j$  using the quantity data values from the current stream and calculate the  $m_i$  before writing to the proper output stream file.

As reproduced above, the conversion procedure has entries for various conditions corresponding to values of PvtNum, PFlag, and the various pressure nodes. If the current value of these variables associated with a particular stream does not match the values specified in the conversion table, linear interpolation is used. Since PvtNum and PFlag are integers they will never be interpolated. Streams will always adhere to one PVT region and gas injection will either be on or off. Pressure, on the other hand, will rarely match and will always be interpolated.

At this point Streamz “knows” how to convert from the BO\_Char to the EOS9 characterization. We return to the parent file and proceed further.

```
DEFINE out_str_file = 'e100-bo-9c.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?
END
INCLUDE 'tabulate.inc'
```

We define another token and then open the **Output** stream file. Since the characterization active at this point is EOS9, this file will contain streams in this characterization. The **End** command signifies the end of the scope of the previous primary keyword (in this case **StreamFile**). We then proceed to look at the included file **tabulate.inc**.

```
;Convert and tabulate output on well basis
DOMAIN TIME T1 T2
TABULATE WELL AND TIME (DAYS)

STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE
```

A commented line (starting with ;) tells the user that the command will convert the streams and tabulate them on a well basis. The **Domain** command creates a named domain (i.e. a slice) called **Time** from the two variables **T1** and **T2**. Usually a domain is used for accurate integration and differentiation when

streams are being combined. In this example it is used merely as a replacement for the two variables, as in the **Tabulate** command.

The **Tabulate** command is one of the Streamz commands initiating a conversion. Upon encountering it Streamz will attempt to step through all open input files and write to all open output files. If the characterizations of the two files are different it will go looking for a converter and perform the needed conversion. In addition to initiating a conversion, **Tabulate** also writes out to the output stream file in a tabular format. All requested variables (**Well**, **T1**, and **T2** in this case through the use of the domain **Time**) would be written as stream headings. All consecutive streams that have the requested variables common will be summed up. This allows the automatic totaling of connection rates to well rates *after* the conversion has been performed at connection level. Refer to section 2.3.5 for a discussion on why conversions are performed on a connection basis.

**StreamFile** commands then instruct Streamz to **Close** both the input and output files. This completes the initial BOz conversion of Eclipse results.

Aggregation, reporting and plotting is an important function required frequently after simulations have been run. This example required the grouping of well rates to group levels. We progress further in the same data set.

```
; Use tabulated well streams as new input file
STREAMFILE INP2: INPUT ?out_str_file?
; Prefix "grp-" to group level output file name
STREAMFILE OUT2: OUTPUT Results\grp-?out_str_file?
```

The first **StreamFile** command reopens the old output stream file, but this time for **Input**. A new output stream file is opened and its name is created by prefixing **grp-** to the name of the current input file. This demonstrates a utility provided by Streamz's **define** command. In fact the file is instructed to be created in a sub-folder (called **Results**) to the current folder. This folder needs to exist; Streamz does not create folders.

Note that both files are opened when EOS9 characterization is active. We will merely be grouping and adding streams and no change of characterization will take place.

```
; Define variable FLOW as production of first component
LUMP FLOW C1
```

The **Lump** command creates a lumped component from existing components. Such a lump is used in **Filters** or when specifying conditions for **Split** commands. In either usage a property of the lump is used. This property (e.g. **moles**) is calculated on-the-fly before each stream is processed to obtain

its value. This allows a compositional property of the current stream to be used just as variables are used.

The actual data set uses multiple groupings, but we select a single one to demonstrate the usage.

```
; Only include production wells
FILTER PROD FLOW MOLES GE 0
FILTER OSC_PROD PROD AND WELL SW 'C'
SET GROUP OSC
TABULATE GROUP AND TIME (DAYS) IF OSC_PROD
```

The **Filter** command creates a named (called **Prod**) filter in memory. The condition for which this filter is true is then listed. In our example the condition is: **Flow Moles GE 0**. The meaning is quite intuitive. **Production** is true if the **moles** property of the **lump Flow** is **greater than or equal** to **0**. If this filter is used in a stream conversion command, Streamz will evaluate the value of the lump (in this case the moles of component C1) in the stream, evaluate its **moles** property and compare the value with 0. It will select the stream only if the evaluation returns true and then convert the stream as required.

A further **Filter** is defined refining the selection to those with the variable **Well** starting with (**SW**) **'C'**. The user knows that well names beginning with C belong to the group OSC. The **Set Group** command associates the variable **Group** with a value of OSC to each stream that will be written to the output stream file.

The **Tabulate** command initiates the conversion. This time the variables **Group** and **Time** are requested and will be tabulated as stream headings. The **If OSC\_Prod** option to this command instructs Streamz to select only those streams that satisfy the filter names OSC\_Prod. This is a composite filter selecting streams for which the C1 moles is greater than 0 and for which the well names start with C. The tabulation adds these streams to single group streams at each time step.

### 3.4.2 HTHP Rich Gas Condensate Reservoir: Pressure Depletion

The special feature of this case is the double BO→EOS6→EOS17 conversion. This first conversion uses BOz method and the second one uses the generic conversion matrix, but again using pressure dependence for the split factors. PSM linkers are used to generate the conversion table using the BOz method. PSM linkers also convert results in Eclipse format to stream files in Streamz format. We look at the Streamz data set used:

```
TITLE 'Streamz input file generated by PreStreamz'
INCLUDE 'bo.chr'
```

```

DEFINE inp_str_file = 'crit-gc.str'
STREAMFILE INP1: INPUT ?inp_str_file?
INCLUDE 'crit-gc.cnv'
DEFINE out_str_file = 'Results\crit-gc-6c.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?
END
COPY

```

This Streamz driver file is identical to the previous example's except the last **Copy** command. The input stream file is the Eclipse results adapted for Streamz by PSM linkers. The included **bo.chr** file is identical as the black-oil characterization is always the same. We take a brief look at the included **crit-gc.cnv** file.

```

CONVERT 'BO_Char'   from      VOLUMES   to      MOLES
SET      PVTNUM      1          PRES      (BARA)   422.073
SPLIT    SO          X1
          3.6596E-03   8.2778E-01  1.0088E+00  9.3593E-01  1.2752E+00
          2.1465E-01
SPLIT    SG          X1
          3.8424E-02   4.5197E-03  1.0608E-03  2.3661E-04  -5.1486E-04
          -2.3546E-04
SET      PVTNUM      1          PRES      (BARA)   420
SPLIT    SO          X1
          1.9389E-03   8.3220E-01  1.0133E+00  9.3906E-01  1.2764E+00
...

```

Initially the full EOS characterization is defined (not shown in the extract above). The **Convert** command defines the conversion from black-oil volumetric rates to molar rates. The notable omission is the **PFlag** variable in the **Set** command. This variable is only used for gas-injection cases. **PvtNum** is still used indicating that there are multiple PVT regions in the model. In fact this model has 24 regions but all the regions have identical properties. A look at the conversion file indicates that 24 sets of (identical) tables are provided. Although a single table would have sufficed providing identical conversions, use of the 24 tables provides consistency with the PVT tables in the reservoir model.

Each **Split** command lists the split factor of each input component for all the output components. As can be seen, some component split factors are negative. This is discussed in section 2.3.2.

The **Output** stream file is opened when the output EOS characterization is active. The Streamz driver file ends with a simple **Copy** command. This command is the simplest way to initiate a conversion in Streamz. If the relevant characterizations and conversions have been defined and appropriate files have been opened, the **Copy** command would step through each input open file and write to each open output file. Each stream would be written in the same order along with each and every variable. All stream headings would

be reproduced. All **Set** commands in the stream file would be written out. Only the required conversions (if any) would be performed and the stream quantity data would be written out for the new characterization.

The **Copy** command is ideal if all the characteristics of the streams have to be retained while only performing the conversion to the new characterization. This is exactly what is needed in this case since a second conversion is to be performed to a 17-component characterization. We need to retain the original fluid characteristics of composition and pressure. Any aggregation would alter these characteristics resulting in the next conversion not being as accurate as possible.

The second conversion uses a similar Streamz driver file. PSM/Toolkit allows the running of multiple Streamz conversions from a single interface. We take a brief look at this driver file:

```
TITLE 'Streamz input file generated by PSM Toolkit'
INCLUDE '6COMP.chr'
DEFINE inp_str_file = 'Results\crit-gc-6c.str'
STREAMFILE INP1: INPUT ?inp_str_file?
INCLUDE '6c-17c.cnv'
DEFINE out_str_file = 'Results\crit-gc-6c-17c.str'
END
COPY
```

We take a look at the included file **6Comp.chr** that was also a part of the included conversion file in the previous conversion.

```
CHAR 'crit-gc EOS6'
```

COMP	MW	TC K	PC BARA	AF	VS	AMOD	BMOD
X1	18.64	206.95	42.5	2.80E-02	0.00E+00	1	1
X2	58.89	426.65	35.2	0.184	0.00E+00	1	1
X3	112.42	618.15	28	0.332	0.00E+00	1	1
CN1	179.98	698.15	21.3	0.44	0.00E+00	1	1
CN2	310	843.15	17.3	0.54	0.00E+00	1	1
CN3	480	1198.15	15	0.65	0.00E+00	1	1

BIPS	X1	X2	X3	CN1	CN2	CN3
X1	0.00E+00	0.00E+00	1.87E-02	6.50E-02	7.00E-02	7.00E-02
X2	0.00E+00	0.00E+00	2.10E-03	2.15E-02	2.15E-02	2.15E-02
X3	1.87E-02	2.10E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CN1	6.50E-02	2.15E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CN2	7.00E-02	2.15E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CN3	7.00E-02	2.15E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00

This is a complete EOS characterization with 6 components. All 6 are pseudo-components. The input stream file corresponds to this characterization and is the same stream file written to in the previous conversion.

The included conversion file **6c-17c.cnv** contains some new features of Streamz. The first line includes the file **17Comp.chr**.

```
Inlcude 17COMP.chr
```

A look at this file reveals that it defines the 17-component characterization and names it **17Comp\_CND**.

```
CHAR      17COMP_CND
Comp      Mw      Tc      Pc
          K      BARA
N2        28.01   -146.95  33.94
CO2       44.01   31.05   73.76
C1        16.04   -82.55   46
C2        30.07   32.25   48.84
C3        44.1    96.65   42.46
IC4       58.12   134.95  36.48
NC4       58.12   152.05  38
IC5       71.837  193.98  35.247
NC5       72.15   196.45  33.741
C6        84.198  250.8   34.7
C7        97.464  285.07  32.1
C8        111.251 313.35  29.8
C9        125.799 337.1   27.5
C10       139.157 361.88  26
CN1       180     415     21.9
CN2       310     550     16.3
CN3       480     925     15.9
end
```

It is a partially complete EOS characterization. The only property being used in the current conversion is **Mw**. The other properties might be useful if this stream participates in an EOS based conversion (e.g. Streamz' **Process** command). But they are not mandatory for the current conversion. One method for conversion among characterizations is the Gamma Distribution (see section 2.2). The basic data used in GD is the molecular weight. Continuing further in the conversion file.

```
Convert 'crit-gc EOS6' from Moles to Moles conserving Moles
Gamma X3 C7
AVE = 1
Set Pres      423 (bara)
SPLIT X1      CO2  0.03514
SPLIT X1      N2   0.00462
SPLIT X1      C1   0.84342
SPLIT X1      C2   0.11682
SPLIT X2      C3   0.50204
SPLIT X2      IC4  0.07055
SPLIT X2      NC4  0.19427
SPLIT X2      IC5  0.05624
SPLIT X2      NC5  0.08078
SPLIT X2      C6   0.09611
...
```

The **Convert** command defines the conversion from ‘crit-gc EOS6’ characterization to the currently active ‘17Comp\_CND’. It specifies that both the input and the output streams should be in **Moles**. It also specifies that **Moles** should be **Conserved** while performing the conversion. In general it is only possible to conserve either mass or moles when performing conversion using the Gamma Distribution. The conserve command also prompts the issue of warnings if the desired unit is not conserved when split factors are specified.

The **Gamma** command instructs Streamz to use the GD to perform the conversion. Mandatory in the use of GD is the specification of the input and output component names that would participate in the GD conversion. In our example components X3 and subsequent from the input characterization streams and components C7 and subsequent from the output streams are requested for GD conversion. Each stream from the input stream file would be fit to the GD. Streamz would then use the GD-parameters obtained in the fit, together with the molecular weights of the output characterization, to calculate the molar rates of the output stream.

The **Ave** command is a sub-keyword to **Convert** and specifies the average (absolute or relative) molecular weight of the GD. It allows the user to specify the starting value and the upper- and lower limits of this parameter when the input stream is being fit. As used, with a fixed value of 1, it freezes this parameter. The truncated GD (see section 2.2.2) is a 4-parameter probability density function. In this example we have 4 fractions (X3, CN1, CN2, and CN3) that are participating in the GD fit. This is unlikely to give a unique solution. Hence we fix one parameter. Ave is a good choice as we specify that the average molecular weight of the heavier fractions should remain the same as input.

We see that the conversion method in this example is a combination of **Gamma** and **Split**. Since GD is applicable only to heptanes plus, a separate method is specified for the conversion of the lighter components. The example uses a set of split factors that are piece-wise linear functions of the pressure. The pressures are **Set** and at each node pressure two **Split** command specify the splitting of each input component (X1 and X2) into 4 and 6 output components respectively. X1 splits into CO<sub>2</sub>, N<sub>2</sub>, C<sub>1</sub> and C<sub>2</sub> while X2 splits into C<sub>3</sub>, iC<sub>4</sub>, nC<sub>4</sub>, iC<sub>5</sub>, nC<sub>5</sub>, and C<sub>6</sub>. At each pressure the split is slightly different. For this example, the split factors were obtained by performing a CVD experiment in a PVT program and calculating the split factor by trivial lumping (see section 2.4.3).

Hence, in our example, Streamz would fit the X3 and higher components of each stream to the GD and obtain C<sub>7</sub> and higher component molar amounts.

For the same stream Streamz would calculate interpolated split factors for X1 and X2 for the actual pressure associated with the stream and use those to obtain the molar amounts of the pure components CO<sub>2</sub> to C<sub>6</sub>. Thus attributes of each connection stream would be used to perform the conversion providing maximum accuracy.

Returning to the main Streamz driver file, the output stream file is opened while the 17-component characterization is active. The **Copy** command actually initiates the conversion to the 17-component characterization. As before, each stream is merely converted without any other manipulation and the format of the stream file is kept intact.

A further step in the example was the separate tabulation of all stream files. Only the tabulation of the 17-component stream file is discussed, the other implementations are identical except for the input and output stream files. We look at the Streamz driver file implementing the tabulation:

```
TITLE 'Streamz input file generated by PreStreamz'
INCLUDE '17COMP.chr'
DEFINE inp_str_file = 'Results\crit-gc-6c-17c.str'
STREAMFILE INP1: INPUT ?inp_str_file?
DEFINE out_str_file = 'Results\crit-gc-6c-17c-tab.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?
INCLUDE 'tab-17c.inc'
```

It follows a similar structure. The 17-component characterization file is **included**. Both the input and output stream files are opened when the same characterization is active. Also notable is the absence of any **Convert** command (or conversion file). We are keeping the same characterization. An include file actually performs the tabulation. This structure of the Streamz driver file is generated automatically by the PSM/Tk to allow multi-step, multi-conversion batch runs. If manually generated by the user, the tabulation could have been instructed in the main file. We now look at file performing the tabulation.

```
domain time t1 t2
lump tot 17*1
filter prod tot moles gt 0
tabulate time days if prod
```

The **Domain** command usage is similar to the example in section 3.4.1. The **Lump** command however is slightly different. The lump **tot** is made up of the whole of all the 17 components (**17\*1**). The repetitive operator (**\***) allows the short hand notation to specify 17 1s. In the context of the **Lump** command the absence of a component name implies the next to the one previously used. The first defaults to the first component of the active characterization. Thus the command is syntactically equivalent to

```
Lump tot N2 1 CO2 1 C1 1 C2 1 C3 1 IC4 1 NC4 1 IC5 1 NC5 1 C6 1 C7 1
C8 1 C9 1 C10 1 CN1 1 CN2 1 CN3
```

A **Filter** named **Prod** is defined which would be satisfied only if the **Moles** property of the lump **tot** is greater than (**gt**) zero. In other words the filter screens out any non-producing connections. This filter is used in the **Tabulate** command. Since only the domain **Time** is specified, the associated variables **T1** and **T2** will be tabulated along with the molar quantities. This also means that the consecutive streams that are common (connections and wells) would be summed up giving the field rates at each time step.

Since the variables **T1** and **T2** have been declared to be of type **Time** (not to be confused with the domain of the same name) they can be requested in any of the supported time units. The actual units would be used while tabulating the variables. Similar tabulation include files are used for the 6 component and the black-oil stream files.

### 3.4.3 Rich Gas Condensate with Compositional Gradient

The special feature of this case is compositional gradient resulting in a different initial fluid composition varying with depth. Black oil simulation results are post-processed by the BOz method. Conversions are performed at grid-cell level thereby capturing the correct GOR and converting it to the correct compositions. Compositions (molar rates) are then summed to well-level for reporting.

We need to convert black-oil streams to a 6-component EOS characterization (eos6), the conversion file for which is pre-generated using the BOz method. This conversion is sensitive to the grid-block pressure, so this information must be available in the stream file.

Next, we need to convert our streams from the eos6 characterization to a 22-component characterization (eos22), perhaps for process requirements. The conversion file for this is also pre-generated. Since this conversion is valid for well-based streams, we can improve our efficiency by summing away the connection information during the prior conversion to eos6. After the conversion to eos22, we need to report results on a yearly basis for all wells and for the whole field.

Taking a look at the main Streamz driver file,

```
TITLE 'Streamz input file generated by PSM Toolkit'
INCLUDE 'bo.chr'
STREAMFILE INP1: INPUT 'bo_3w.str'
INCLUDE 'bo-eos6.cnv'
STREAMFILE OUT1: OUTPUT 'eos6_3w_a.str'
INCLUDE 'bo-eos6-eos22_3w.inc'
```

This is an extremely simple and basic, yet complete, input file for Streamz. It follows the same structure as in the previous examples. We concentrate on the file included by the last **Include** command. This is used to introduce the commands to input, convert, and output the desired streams. Without any such command in this data set, Streamz would basically do nothing.

```
TITLE 'Conversion to 6-Component EOS'
DOMAIN TIME T1 T2
FILTER VALID: PRES GT 0 BARA
TABULATE TSTEP and TIME (days) and WELL if VALID
STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE

STREAMFILE INP1 INPUT 'eos6_3w_a.str'
STREAMFILE OUT1 OUTPT 'eos6_3w_b.str'

FILTER WELL: WELL EQ 'P5010', COPY IF WELL
FILTER WELL: WELL EQ 'P2505', COPY IF WELL
FILTER WELL: WELL EQ 'P0101', COPY IF WELL

TABULATE TSTEP AND TIME (days)

STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE

STREAMFILE INP1 INPUT 'eos6_3w_b.str'

LUMP C7PLUS = C7-9F1-2, F3-8, F9

INCLUDE 'eos6-eos22.cnv'

STREAMFILE OUT1 OUTPT 'eos22_3w_a.str'

COPY

STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE

STREAMFILE OUT1 OUTPT 'eos22_3w_b.str', VARIABLE LABEL STRING

LUMP TOTAL 22*1

DEFINE INFILE 'eos22_3w_a.str'

DEFINE NAME 'P5010', INCLUDE 'wellsums.inc'
DEFINE NAME 'P2505', INCLUDE 'wellsums.inc'
DEFINE NAME 'P0101', INCLUDE 'wellsums.inc'
DEFINE NAME '', INCLUDE 'wellsums.inc'
```

We concentrate on new commands only. The **Filter** command creates a filter named **Valid**, requiring the variable **Press** to be greater than 0 bara. This will filter out the streams from the un-drilled well.

The **Tabulate** command triggers a read/conversion/write operation from the input stream file to the output stream file. Only the specified variables and domains are retained (along with their specified units, as requested) in the

output stream file. All connection streams are summed to well streams and the streams are tabulated.

The data set then uses a pair of **StreamFile** commands to close both input and output stream files, re-open the previous output stream file for *input*, and then open a new output stream file ('eos6\_3w\_b.str'). We then define a **Filter** called **Well**, which holds true only for streams having variable **Well** equal to 'P5010' (note that filter names will not conflict with variable names). We then perform a **Copy If Well** command. All streams for which the filter applies (i.e., where variable **Well** equals 'P5010') will be copied from the input stream file to the output stream file.

This is repeated for the other two wells ('P2505' and 'P0101'), noting that filter **Well** is redefined each time. As a result, three stream tables (one table for each well) will be written to the output stream file. The streams in each table will be ordered by time step. Thus we achieve sorting of streams by well.

Another **Tabulate** command is then issued, this time without specifying the **Well** variable (or any filters). The result is the summing of well streams into composite field streams for each time-step. The output stream file now contains a fourth stream table, this one listing total field rates by time-step.

Both input and output stream files are again closed and the output file ('eos6\_3w\_b.str') is again opened as input for the next stage, which will convert the streams from the eos6 characterization to eos22. We skip this conversion in this example description, which is covered in the next example (section 3.4.4). The unique features of this conversion are the sorting of streams to well basis and the aggregation into yearly averages.

Instructions will now be given to obtain yearly-averaged daily stream rates for each well and the field as a whole. A new output stream file ('eos22\_3w\_b.str') is opened to which streams corresponding to the "current" (eos22) characterization will be written. Additionally, the **Variable** sub-command defines a new string variable called **Label** and makes it the file's first variable. We then define a new lumped fraction named **Total** that will **Lump** the entire amounts of all 22 components. We plan to use this lumped fraction in a subsequent filter operation.

The **Define** command is then used to associate the token **Infile** with the character string 'eos22\_3w\_a.str'. This command allows a run-time replacement of any occurrence of the token surrounded by question marks (in this case, ?INFILE?) with its associated string (in this case, 'eos22\_3w\_a.str', but without the quotes). All replacements will occur before any other parsing of an affected input line. This definition will persist throughout the rest of the

driver file in which it is issued; carrying over into included files as well. It will not carry back into a parent driver file, however.

The next line **Defines** the token **Name**, associates it with the replacement string 'P5010' (the name of a well) and then **Includes** the file 'wellsums.inc'. The next three lines redefine **Name** and **Includes** the same file. Hence we are trying to execute the same set of operations on each well name and then on a blank string, the reasons for which will become clear as we go along. We now look at the contents of this **Included** file.

```

STREAMFILE INP1 INPUT '?INFILE?'
DEFINE TEMPFILE 'streamz_temp.str'
STREAMFILE TMP1 OUTPUT '?TEMPFILE?'

FILTER WELL: WELL EQ '?NAME?', COPY TO TMP1 IF WELL

STREAMFILE INP1 CLOSE
STREAMFILE TMP1 CLOSE
STREAMFILE INP1 INPUT '?TEMPFILE?'

DEFINE T2 '0.0'
DEFINE T1 '?T2?', DEFINE T2 '1.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '2.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '3.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '4.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '5.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '6.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '7.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '8.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '9.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '10.0', INCLUDE timesums.inc

WRITE STREAMS YEARS_0.0-1.0
      AND YEARS_1.0-2.0
      AND YEARS_2.0-3.0
      AND YEARS_3.0-4.0
      AND YEARS_4.0-5.0
      AND YEARS_5.0-6.0
      AND YEARS_6.0-7.0
      AND YEARS_7.0-8.0
      AND YEARS_8.0-9.0
      AND YEARS_9.0-10.0

STREAMFILE INP1 CLOSE
CLEAR

```

This is designed to be a generic include file that can be used with many data sets. It illustrates many unique features of Streamz. The first command opens a **StreamFile** with the nickname INP1 for input. The actual name is derived from the defined wildcard ?INFILE?, which actually becomes 'eos22\_3w\_a.str' (without the quotes). If this include file is used in another data set it can easily take the name of any other stream file. Another definition assigns the name of a temporary stream file, 'streamz\_temp.str', to the token **Tempfile**. That file is opened for output.

The **Filter** command creates a filter named **Well**, which holds true whenever the variable **Well** equals `?NAME?`. The wildcard `?NAME?` gets replaced by its current value (a well name in the first 3 cases and an empty string in the last case). The **Copy** command applies the filter (by its use of **If Well** option) and writes all the streams with that particular well name **To** the temporary file `TMP1`. Note that when the well name is empty, the filter will select the field-rate streams, since a previous **Tabulate** command has disassociated them from any well name.

Both stream files are closed and the temporary file is reopened for input. A series of definitions for tokens `T1` and `T2` (which will represent the start and end times for each of ten different yearly time intervals) are made and the file `'timesums.inc'` is included at each step. This, in a sense, is the inner loop, the outer loop being to **Define** the `NAME` token and to **Include** `'wellsums.inc'` for each well and then for the field (by assigning the empty string to `NAME`). We end up executing the contents of `'timesums.inc'` for each of ten yearly time intervals for each well, and then for the field. This is analogous to a nested “time loop” executing within an outer “well loop”.

```
FILTER ACTIVE: TOTAL MOLES NE 0.0
FILTER YEAR: TIME GE ?T1? YEARS AND TIME LE ?T2? YEARS AND ACTIVE
COMBINE YEARS_?T1?-?T2? IF YEAR, WEIGHTING OVER TIME (days)
TAG YEARS_?T1?-?T2?: LABEL = 'Years ?T1? to ?T2?'
```

The file `'timesums.inc'` is also intended to be generic. It first creates a filter called **ACTIVE**, which allows only those streams where the **Moles** property of the lumped fraction **TOTAL** is non-zero. In effect, only the streams with non-zero total molar rates will be counted, thus excluding streams from the inactive time-steps of any well. Then a **Filter** called **YEAR** is created, adding the **ACTIVE** requirement to the additional requirement that a stream's **TIME** domain must intersect some part of the time interval between the current values (in **YEARS**) of the `?T1?` and `?T2?` wildcards. Only the portion of the stream that intersects this interval will be selected. This could be the entire stream or a fractional portion, and is the primary reason for **Domain** definitions. The intersection is determined by the values of the stream's **TIME** domain variables, `T1` and `T2` (not to be confused with the defined tokens of the same names). If the value of either the `T1` or `T2` time variable falls between `?T1?` years and `?T2?` years, at least some portion of the stream will be selected. The units of **Years** can be used here even though the input streams have their time variables, `T1` and `T2`, given in **Days**. Streamz internally converts the known units.

The **Combine** command is used to create a named stream. The name will be a concatenation of the strings `'YEARS_'`, the current value of `?T1?`, `'-'`, and the current value of `?T2?`. It selects and combines only the portions of the streams that satisfy the filter **YEAR**. For example, if `T1` has been **Defined** as `'0.0'` and

T2 has been **Defined** as '1.0', the portions of all streams that fall within the 0.0 to 1.0 year time interval (0 to 365.25 days) will be combined into a stream named YEARS\_0.0-1.0. Now suppose a stream has its T1 and T2 time variables equaling 360.25 and 370.25 days, respectively. Half of that stream will contribute to the combined stream.

The **'Weight Over'** option of the **Combine** command is used to correctly obtain time-averaged stream rates. Since the input streams are in daily rates, we need to multiply the rates at each chosen time interval by the time interval itself (in days), combine the resulting cumulative amounts, and then divide by the sum of the chosen time intervals (in days, for average daily rates, hours for average hourly rates, etc.). The multiplication can be achieved by the **'Weigh Time Days'** option, where TIME is defined as the **Domain** between the beginning and ending time variables for each stream. The division can be achieved by the **'Over Time Days'** option (for average hourly rates, substitute **Hours**, for average yearly rates, substitute **Years**, etc.). The **Combine** command will handle the accumulation of both the numerator and denominator, correctly accounting for any partially selected streams. The **'Weight Over Time Days'** option is a short cut for the combination of the **'Weigh Time Days'** option and the **'Over Time Days'** option. Note that, for cosmetic purposes, the driver file also puts parentheses around the required time units. These are optional and the program simply ignores them.

The **Tag** command allows a variable of a given value to be attached to a named stream. The name of the stream serves as the command's first argument. In this example, it's the name of the stream just created by the **Combine** command. The next argument is the name of a previously declared variable (LABEL, in this case), followed by the desired value for the variable (including its units, if required). Additional variables and their values may follow. In this example, LABEL will be assigned the character string 'Years ?T1? to ?T2?' to indicate the corresponding time interval. The actual values of the ?T1? and ?T2? wildcards will first be substituted, of course.

Note that no output streams are written within the 'timesums.inc' driver file. It merely creates a named stream for a time interval of interest. The output is done in the parent file ('wellsums.inc'). After 'timesums.inc' has been included 10 times, creating 10 named streams, those streams are written to the output stream file by a **Write** command. The first argument to **Write** is the sub-keyword **Streams**. Then follow the names of all the streams we wish to write, in order, separated by the keyword **And**. These are the streams created within the 'timesums.inc' file.

The temporary input file is closed, and the **Clear** command clears all the filters and named streams from memory, allowing further processing to be more efficient. It is shorthand for **Clear Filters, Clear Streams**.

Recall that the commands in 'wellsums.inc' apply to a single given well (or the field as a whole). They end up writing a table of daily-rate streams, time-averaged on a yearly basis, for the well of interest (or the entire field).

After the file 'wellsums.inc' is executed for each well and lastly for the field (achieved by assigning an empty string to NAME), the control passes back recursively to each parent file, ending with the main driver file 'bo-eos6-eos22\_3w.stz'. Since no further commands follow, Streamz ends its run with some statistics on errors, warnings and execution time.

This example provides excellent insight into the power provided by imaginative use of the **Define** and **Include** commands. New commands **Combine**, **Tag** and **Write** were also discussed.

#### 3.4.4 Gas Injection in Rich GC reservoir with Constant Composition

This example is from a North Sea 3D Sector model containing a rich gas condensate fluid with constant composition. Gas injection takes place from a single up-dip well and production from one down-dip well. Black oil simulation results are post-processed by the BOz method. Conversions are performed at grid-cell level thereby capturing the correct GOR and converting it to the correct compositions. Compositions (molar rates) are then summed to well-level. Conversions from 6-component to 22-component molar rates are performed at well level with split factors a function of the well stream  $C_{7+}$  mole fraction.

The conversion from black-oil to compositional is identical to the full field example discussed in section 3.4.1. The split factors are functions of PvtNum, PFlag, and Pressure. The Streamz driver file is identical and uses a common tabulation include file, the content of which are reproduced below.

```
domain time t1 t2
filter prod well sw 'P'
tabulate well and time days if prod
```

The commands are similar to those discussed in the previous discussions except that the **Filter** selects only the production wells (**well sw 'P'**). We proceed to the conversion from 6 components to 22 components. The necessary converter needs to use the mole fraction of the  $C_{7+}$  components as the control variable for splitting the input components into output components. The implication is that this splitting depends on the amount of  $C_{7+}$  components present in the input stream. We look at the driver file.

```
TITLE 'Streamz input file generated by PreStreamz'
INCLUDE 'eos6.chr'
LUMP C7PLUS C7-9F1-2 F3-8 F9
```

```

DEFINE inp_str_file = 'bo6-eos6.str'
STREAMFILE INP1: INPUT ?inp_str_file?
INCLUDE 'eos6-eos22.cnv'
DEFINE out_str_file = 'Results\bo6-eos6-eos22.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?
END
INCLUDE 'tabulate.inc'

```

Most of the commands have already been discussed. The **Lump** command combines the entire portions of the last 3 components into a lumped fraction named **C7Plus** (absence of amounts defaults to 1). This is used in the included conversion file, a portion of which we now look at.

```

Include 'eos22.chr'
Convert 6COMP from MOLES to MOLES
Set C7PLUS MOLES/MOLE 0.134427371468373
Split C1N2      N2  0.00276744204231122
Split CO2C2     CO2 0.380573441154535
Split C1N2      C1  0.997232557957689
Split CO2C2     C2  0.619426558845464
Split C3-6      C3  0.506522522490617
Split C3-6      IC4 0.089045004254204
Split C3-6      NC4 0.17069738842343
Split C3-6      IC5 0.0624166091913825
Split C3-6      NC5 0.0693791606867286
Split C3-6      C6  0.101939314953638
Split C7-9F1-2 C7  0.232139503801378
Split C7-9F1-2 C8  0.275194684890061
Split C7-9F1-2 C9  0.174448467122042
Split C7-9F1-2 F1  0.174174654613321
Split C7-9F1-2 F2  0.144042689573199
Split F3-8      F3  0.183156820273156
Split F3-8      F4  0.220960673848973
Split F3-8      F5  0.117507342080668
Split F3-8      F6  0.183070289681187
Split F3-8      F7  0.131345025913108
Split F3-8      F8  0.163959848202908
Split F9        F9  1
...

```

The file initially **Includes** the 22-component characterization. The **Convert** command specifies conversion from molar to molar units. We then encounter a new type of **Set** command where the **Moles/Mole** property of the previously defined **Lump C7Plus** is set at a particular value. **Split** commands specify the amount of each input component that goes into each output component. These split factors were actually generated from the 22-component simulation assuming the trivial lumping based on the original pseudoization.

The split factors cover a range of values of mole fraction (= **Moles/Mole**) of  $C_{7+}$  (= **Lump C7Plus**).

A new output stream file is opened to which streams corresponding to the “current” (eos22) characterization will be written. The **Copy** command

initiates the read/write operation, performing the necessary conversions on the fly. The **C7Plus** lumped fraction is calculated for each stream prior to conversion. The value of the **C7Plus Moles/Mole** property (i.e., the mole fraction of  $C_{7+}$  in the stream) determines, through linear interpolation, the set of split factors used for the conversion.

The same tabulation include file is used to effect the conversion. This example has a single production well so the tabulated stream files are already sorted. The output stream file now contains the molar rates corresponding to the 'eos22' characterization.

### 3.4.5 TBP Stream Data Fit

Streamz always converts from one characterization to another. Frequently we need only to fit measured data to the GD. We can either convert to a dummy single component characterization or to a characterization that we might use in future. In absence of an available characterization to convert to, it is recommended to convert to a single carbon number (SCN) characterization. This application was described in 2.2.5. In this section we look at the data set that achieved it.

```
ECHO ON
title 'FIT TBP and split to K-F SCN'
```

```
char 'TBP'
comp      MW
C7S       96
C8S       101
C9S       114
C10S      129
C11S      144
C12S      163
C13S      177
C14S      192
C15+     330
END
```

```
streamfile in input tbp.str
splitfile spl1 open GetSplits.spl
gammafile gam1 open GetSplits.gam
```

```
CHAR 'SCN'
COMPS   MW   LMW
C7      96
C8      107
C9      121
C10     134
C11     147
C12     161
C13     175
C14     190
C15     206
C16     222
C17     237
C18     256
```

```

C19      263
C20      275
C21      291
C22      305
C23      318
C24      331
C25      345
C26      359
C27      374
C28      388
C29      402
C30      416
C31      430
C32      444
C33      458
C34      472
C35      486
C36      500
C37      514
C38      528
C39      542
C40      556
C41      570
C42      584
C43      598
C44      612
C45              619
END

```

```
streamfile out output tbp-scn.str
```

```

convert TBP from MOLES to MOLES conserving MASS
splits spl1
gamma C7S C7 file gam1

```

```
copy
```

A new Streamz command, **Echo On**, is introduced. This turns on echoing of read commands from the driver file and is frequently used to debug a long data set. It instructs Streamz to write to the standard output (the screen and/or a file) all input commands along with line numbers. This command can also be used to understand Streamz's interpretations of the commands in the driver file.

**Title** and **Char** commands have been discussed in previous examples. It is notable that the characterization only contains  $C_{7+}$  component. The **StreamFile** command opens the input stream file. In this case there is a single stream (the measured TBP data) in the stream file. We look at the contents of this stream file.

```

STREAMZ 1
Char    'TBP'
DATA

MOLES
1.12   1.3    1.18   0.98   0.62   0.57   0.74   0.53   4.1

```

This is a bare-bones stream file containing only **molar** quantity data. Proceeding further in the main driver file, the **SplitFile** and **GammaFile** are new commands. They specify the opening of files that can contain split factor and gamma fit related output respectively. Nicknames are given to each for use later in the data set.

Another **Char** command defines the SCN characterization. The MW data is taken from Katz-Firoozabadi<sup>6</sup> table. An important point to note is the inclusion of the LMW property of the characterization. This specifies the lower molecular weight bound for the fraction and comes into play when a regressed GD model is split into the output characterization. As explained in section 2.2.4, Streamz will discretize a model by fixing molecular weight bounds. In absence of LMW in the output characterization it calculates them as averages of individual fraction molecular weights. For example, if fraction 2 has an MW of 90 and fraction 3 has an MW of 110, the bound would be calculated as 100. This is fine for finely divided fractions such as the SCN.

The problem comes in the last fraction where fraction molecular weight would correspond to C<sub>45+</sub>. Averaging this with the average of the previous fraction would give a very large bound value. The solution is to provide the LMW for this last fraction. Since SCN fractions have an average molecular weight difference of 14, the bound for the next fraction is usually about 7 added to the current fraction's average. So the LMW supplied for the C<sub>45+</sub> fraction in the SCN characterization is calculated as  $612 + 7 = 619$ .

A **StreamFile** is opened to contain the output stream molar amounts. The **Convert** command requests conversion from **Moles** to **Moles** while **Mass** will be conserved. This ensures that the mass of the input and output streams will be identical. The **Splits** command instructs that the split factors for the current convert command will be written out to the file pointed to by the nickname **Spl1**. The convert method to be used is **Gamma** with the first component of both characterizations participating in the GD fit and split. The GD distribution results are to be written to a file pointed to by the nickname **gam1**. Full regression will be used with default limits on all 4 GD parameters.

The **Copy** command initiates the conversion.

### 3.4.6 Splitting of a Single C<sub>7+</sub> Fraction

This example is basically the last half of the previous example. We have a single C<sub>7+</sub> fraction with an average molecular weight and a normalized mole fraction of 1. We have a pre-determined GD model using which we want to split the single fraction into multiple fractions corresponding to the output characterization.

```

title 'Split C7+'

char 'C7+'
comp      MW
C7+      203.779
END

streamfile in input C7+.str

splitfile spll open C7+.spl
gammafile gaml open C7+.gam

char 'TBP'
comp      MW      ~LMW      LMW
C7S      92.91    90.27
C8S      102.25   96.58
C9S      115.68   108.65
C10S     130.72   123.33
C11S     144.25   138.61
C12S     156.33   150.15
C13S     172.00   162.79
C14S     189.49   181.78
C15+     334.48   197.57    197.57
END

streamfile out output tbp.str

convert C7+ from MOLES to MOLES ;conserving MASS
splits spll
gamma C7+ C7s file gaml
Shape      0.72872
Average 203.779
Bound      90.267
Origin     1.00000

copy

```

Streamz always works on two characterizations. The first **Char** command defines the single component characterization with a molecular weight of 203.779. The input stream file, a split file and a gamma file are opened.

Another **Char** command defines a 9-component characterization. The definition illustrates a subtle feature of the tabular input of the characterization. While headings like **Comp**, **MW**, and **LMW** allow the corresponding property data in lined-up columns (see Appendix B), a tilde (~) in front of a heading is like a column comment. It essentially disables the reading of data lined-up under the disabled heading.

The output stream file is opened and the **Convert** command instructs the use of all components of both characterizations to be used in the conversion using the **Gamma** method. The GD conversion will use a fixed model specified by the fixed values of all four parameters. The **Copy** command actually initiates the conversion. The molar amount of the split would be found in the output stream file.

### 3.4.7 Typical Gamma Fit usage for Multiple Samples

Section 2.2.9 describes this example where 5 samples are fit to the GD. We take that data set for discussion but use only two samples for brevity. This reduced data set (driver file) is presented below.

```
GAMMAFILE GAM OPEN 'InitialFitAllSamples.gam'

CHAR 'DST30-6-2'
NAMES MW
FRC1 96
FRC2 107
FRC3 121
FRC4 134
FRC5 147
FRC6 161
FRC7 175
FRC8 190
FRC9 206
FRC10 222
FRC11 237
FRC12 264

Mix DST30-6-2 FRC1 MOLES
0.88 0.77 0.49 0.34 0.213 0.091 0.078 0.041 0.024
      0.012 0.009 0.012

CHAR '30-6-4'
NAMES MW
FRC1 90
FRC2 99
FRC3 106
FRC4 120
FRC5 139
FRC6 146
FRC7 160
FRC8 174
FRC9 194
FRC10 205
FRC11 218
FRC12 234
FRC13 248
FRC14 265
FRC15 470

Mix '30-6-4' FRC1 Mass
2.03 4.76 4.01 4.62 2.38 3.37 3.51 4.05 3.65
      3.93 3.04 3.42 3.15 3 45.87

CHAR DUM
COMP MW
DUM 100

STREAMFILE OUT: OUTPUT 'streamz output/All_out.STR'

CONVERT 'DST30-6-2' from MOLES to MOLES conserving MOLES
GAMMA FRC1 DUM
FILE GAM
ZERO 1.0
CONVERT '30-6-4' from MASS to MASS conserving MASS
```

```

GAMMA FRC1 DUM
FILE GAM
ZERO      1.0

Mix "Sample1a" 1 Stream "DST30-6-2"
Mix "Sample1a" 1 Stream "30-6-4"
Write
Clear

```

A **GammaFile** is opened, as the aim of the task is to obtain GD parameters. The **Characterization** 'DST30-6-2' is defined. A **Mix** command is used instead of the normal stream file to read in a stream. The **Mix** command allows the generation of a named stream corresponding to the active characterization. The mixture is composed of a number of ingredients. Each ingredient has three components: a component name or a previously created stream name, an amount and a unit.

The first mixture in the example is the stream named 'DST30-6-2'. The first ingredient is the first component of the active characterization, FRC1. 0.88 moles of this component are added. The next item on the line is the number 0.77. In absence of an explicit name it defaults to the component next to the one used, i.e. FRC2. In absence of a unit it defaults to the one previously used, i.e. moles. So 0.77 moles of FRC2 is the next ingredient. The succeeding numbers are interpreted in likewise manner. Hence an entirely intuitive method is used to input the measured mole fractions of the sample.

A second **Characterization** is defined and the measured data is input in similar manner. Notable difference is the used of **Mass** instead of moles. Also, from the values of the numbers, it seems the reported amounts are in mass percent. This is irrelevant to the program and readily accepted.

A dummy **Char** is defined with a single component and a dummy molecular weight. This is required since Streamz always needs to convert to *some* characterization. Our aim is to fit the measured data so the converted amounts corresponding to this dummy characterization is thrown away. The output **StreamFile** stores this useless stream.

**Convert** command specifies the method **Gamma** for both the input streams. While first uses the units **Moles** for input, output and conservation, the other uses **Mass**. This is consistent with the way the input streams were prepared. The Gamma Distribution results, which are of utmost important in this task, are written to the gamma file (**File Gam** command) for each conversion and the **Zero** parameter is fixed at 1. This parameter (alias **Origin**) specifies the origin of the GD function (value at which it goes to zero). Fixing it at 1 forces the use of the Classical GD instead of the default Truncated GD. This is consistent with the future use of the gamma parameters in PVTx, which cannot handle the Truncated GD.

The two **Mix** commands towards the end of the data set create streams, both called 'Sample1a', in the active (dummy) characterization. Each specifies the ingredients as 1 stream of the previously created streams. Since this involves a conversion, Streamz will initiate one when each **Mix** command is executed. The amounts of these dummy streams are not significant, only the values written to the GD file are. Hence there is no reason for the **Write** command in this example.

This example actually works in two steps, the first of which was just discussed. The values of **eta** determined are averaged and the GD fit re-calculated fixing this average **eta** (=90). But the second run is identical with the minor alteration in the specification of the **Gamma** command, as shown under:

```
GAMMA FRC1 DUM
FILE GAM
ZERO 1.0 ;
BOUND 90
```

### 3.4.8 Gamma Distribution Application to Reservoir Streams

This example uses Streamz driver files with commands similar to other GD examples discussed in this section. The difference is in the specifications of the **Gamma** method in the **Convert** commands. In this section we only look at this portion of the conversion files that are included in the main data set.

The examples with LMW are not discussed separately as they use the same conversion commands. The only difference is the use of LMW in addition to MW in the **Characterization** command. Example headings that follow use the same nomenclature as the figures in section 2.2.10.

#### Sector Model Split Factors (correct)

The use of split factors computed from 16-component run based on a trivial lumping of components is assumed to give the correct results and other cases are compared with it. The **Convert** command looks like this:

```
Restore 16COMP
Convert 9COMP from MOLES to MOLES

Set C6PLUS MOLES/MOLE 0.386065788909694
Split C1 N2 0.00716733046958036
Split CO2 CO2 1
Split C1 C1 0.99283266953042
Split C2 C2 1
Split C3 C3 1
Split C4 IC4 0.281837288042795
Split C4 C4 0.718162711957205
Split C5 IC5 0.41619606621868
Split C5 C5 0.58380393378132
Split C6F3 C6 0.165511466343883
```

```

Split C6F3 C7+1 0.2962164187349
Split C6F3 C7+2 0.262330542599913
Split C6F3 C7+3 0.275941572321304
Split F4 C7+4 1
Split F5F6 C7+5 0.746636021432688
Split F5F6 C7+6 0.253363978567312
...

```

The split factors are piece-wise linear functions of the mole-fraction of  $C_{6+}$ . This is specified by using the **Set** command with the **Moles/Mole** property of the **Lump** C6Plus. Some input components (e.g. C1) **Split** into more than one output components (e.g. N2 and C1). Some input components (e.g. F4) splits into a single output component (C7+4). The ellipses (...) denote that many such sets of split factors, at different values of **Moles/Mole** property of the **Lump** C6Plus, follow.

### Unrestricted fit of 3 fractions

This conversion tries to fit the 3 input plus fractions to the Truncated GD model. The relevant **Gamma** command is

```
gamma c6f3 c6 file gamfil
```

Unrestricted fit means allowing all the parameters to be regressed. No restrictions on the **Gamma** parameters are specified.

### Fit of 3 fractions while fixing Ave and Origin

This conversion tries to fit the 3 input plus fractions to the Truncated GD model while fixing the values of Ave and Origin. The relevant **Gamma** command is

```

gamma c6f3 c6 file gamfil
ave 1
zero 1

```

The values of both **Origin** and **Average** are fixed at 1.

### Apply fixed Gamma Model (= initial 16c fit)

This conversion applies a fixed GD model to the 9c reservoir streams. This model is obtained from the fit of the 16-component fit using the initial feed. The relevant **Gamma** command is

```

gamma c6f3 c6 file gamfil
; Parameters from fit of initial 16comp stream
Shape 0.75074
Average 1.00000
Bound 84.009
Origin 0.97621

```

The values of **Shape**, **Bound**, **Origin** and **Average** are fixed at values obtained previously. This fixed Gamma distribution would be applied to all reservoir streams.

#### Apply fixed Gamma Model (= average of 16c fit)

This conversion applies a fixed GD model to the reservoir streams. The 16-component reservoir streams are fit to the GD. An average of the parameters are then applied to the 9-component reservoir streams. The relevant **Gamma** command is

```
gamma c6f3 c6 file gamfil
; Parameters from fit of initial 16comp stream (averages caculated
; in spreadsheet "2d-16c-tab.xls")
Shape 0.41828269
Average 1
Bound 84.7484883
Origin 0.964624708
```

The values of **Shape**, **Bound**, **Origin** and **Average** are fixed at values obtained previously. This fixed Gamma distribution would be applied to all reservoir streams.

#### Gamma Model (16c based) function of C<sub>6+</sub>

This conversion applies a varying GD model to the reservoir streams. The 16-component reservoir streams initially are fit to the GD. The parameters are made a piece-wise linear function of the amount of C<sub>6+</sub> in the stream. This linearly varying model is then applied to the 9-component reservoir streams. The relevant **Gamma** command is

```
gamma c6f3 c6 file gamfil
; Parameters from fit of initial 16comp stream

Set C6plus Moles/Mole 0.386065789
Shape 0.75074
Average 1
Bound 84.009
Origin 0.97621

...

Set C6plus Moles/Mole 00.056149675
Shape 0.2
Average 1
Bound 83.791
Origin 0.90383
```

The values of **Shape**, **Bound**, **Origin** and **Average** are specified at particular values of C<sub>6+</sub> mole fraction in the stream. Ellipses (...) denote that other set of gamma parameters at intermediate values of C<sub>6+</sub> **Moles/Mole** are not shown. This partially varying Gamma distribution would be applied to all reservoir streams.

### 3.4.9 Streamz Process command based Well Test Conversions

The conversion of measured well test data (Test Stream) to a corresponding stream at a common set of conditions is performed in a step-wise manner with a number of intermediate streams and conversions. Familiarity with discussions in sections 2.4.1 and 2.4.2 is assumed. Also some background description of the **Process** command is needed.

A **Process** command in Streamz is a combination of any number of “Stages”. A **Stage** is a previously defined “Separator”, which is a stream-splitter. A Stage can take feed from one or more of the previous stages. Any stream from such a previous stage can be **Scaled** and/or **Normalized** before being combined to make the final feed. Each **Separator** is linked to a characterization. If a requested feed does not correspond to the particular characterization it will automatically use a conversion to convert it. This conversion must be previously defined.

A Separator can use one of three different methods for splitting the incoming feed into the vapor and liquid products. The **EOS method** required a pressure and a temperature for flashing the feed at those conditions. The **K-value method** requires a set of K-values for the partitioning of each component into the vapor and the liquid products. These K-values may have dependence on any variable being carried by the incoming stream. The third method is a generic GP Table type **splitting** which are frequently used for approximating the full-blown process simulator. These tables can also be piece-wise functions of any variable associated with the feed stream.

A **Reducer** is a special converter to convert from a multi-component molar stream to a single component volume stream. It uses EOS calculations for this conversion.

Before a **Process** command can be used all separators taking part in the Process must be defined. The characterization associated with a separator is the characterization active when the separator is defined. A conversion should be defined for characterizations of two adjacent separators (if they use different characterizations). A **Process** describes the sequence of named Stages, specifying which previous Stage each feed comes from, which of the products (vapor, liquid or all) of that stage are used, and in what amounts.

For the current example, the initial feed to the Process comes from the Input Stream file containing the reported well-test data.

The first Stage “SP1” is the Test separator. It takes feed from the input stream file. All streams existing in this file will be fed into this stage. The contents of this stream file are described in Table 3.13.

The T\_SC and P\_SC are set at 60F and 1atm respectively. Any other standard followed by the user can be used. Qo\_Test/Qo\_Data is also set at 1 indicating that the Qo\_Data reported in the file is basically the Qo\_Test. T\_Sep\_Ref and P\_Sep\_Ref are set at the required common conditions of 110F and 600psia. T\_Oil\_Ref and P\_Oil\_Ref are set at 60F and 14.7 psia specifying the stock tank conditions of the common separator system.

**Table 3.13 – Description of contents of a typical input stream file used in GOR correction using PSM/Streamz.**

Item	Type	Purpose
STREAMZ	Keyword	Stream file version indicator.
Char	Keyword	Characterization for the stream file.
Well	String Variable	Name of well (for filtering purpose etc.)
Time	String Variable	Time of testing (string)
T_Sep_Test	Temperature Variable	Actual Test separator temperature. Simulated Separator Test also performed at this value.
P_Sep_Test	Pressure Variable	Actual Test separator pressure. Simulated Separator Test also performed at this value.
T_Oil_Test	Temperature Variable	Actual Stock Tank Oil temperature.
P_Oil_Test	Pressure Variable	Actual Stock Tank Oil pressure.
Qg_Data	Volume Variable	Measured Gas Rate.
Qo_Data	Volume Variable	Measured Oil Rate.
GOR_Data	Real Variable	Measured GOR.
Qo_Test / Qo_Data	Real Variable	Test shrinkage factor. Usually calculated once and used to convert actual separator rates to reported well test oil rate.
T_Sep_Ref	Temperature Variable	Common condition Test separator temperature.
P_Sep_Ref	Pressure Variable	Common condition Test separator pressure.
T_Oil_Ref	Temperature Variable	Stock Tank Oil temperature for the Reference Separator system. Usually same as that for the Test Separator.
P_Oil_Ref	Pressure Variable	Stock Tank Oil pressure for the Reference Separator system. Usually same as that for the Test Separator.
T_SC	Temperature Variable	Standard Conditions temperature. Always used for reporting Gas rates.
P_SC	Pressure Variable	Standard Conditions pressure. Always used for reporting Gas rates.
Moles	Quantity Data	Start of columns of quantity data. Must have entries equal to the number of components in the characterization associated with this stream file. Contains the initial feed composition estimate into the Test separator system. Would be identical for all the streams in the file unless an update is available, which would continue identical till another update.

This input stream file can be prepared in MS Excel and linked to data from user Well Test Reporting format. An Automatic Well Test Conversion Formatter has been prepared and can be customized for different formats.

The actual reported well-test results are Qg\_Data amount of gas & Qo\_Data amount of oil. These exist as variables and not quantity data. The quantity data is an estimate of the wellstream that produces the above data, and is feed into the Test Separator at T\_Sep\_Test and P\_Sep\_Test. In our Process this is represented by the sub-command:

```
STAGE SP1 SEP SEP1
```

For each stream in the input stream file the program feeds it to the Separator SEP1 which is of EOS type. EOS flash calculation at T\_Sep\_Test and P\_Sep\_Test result in a vapor product and a liquid product. The next Stage SG1 is a separator SGAS linked to characterization SGAS containing a single component SGAS. Conversion from the EOS characterization of SEP1 is defined by a **Reducer**, which converts moles to volumes:

```
REDUCE "?CHAR_NAME?" from ?MOL_UNIT? to ?QG_UNIT?, Ideal, Temp  
T_SC, Pres P_SC
```

The SGAS separator is a pass-thru separator passing all of the feed without any separation. It merely facilitates the conversion from the EOS characterization to the single component SGAS.

The third stage in the Process is ST1, an EOS separator ST1 linked to the EOS characterization where an EOS flash calculation at T\_Oil\_Test and P\_Oil\_Test is performed. This basically is the stock-tank condition. The liquid at this condition is converted into oil volumes in the stage TO1 which uses the separator TOIL1 with a single component volume characterization. The conversion from moles to volumes uses the reducer:

```
REDUCE "?CHAR_NAME?" from ?MOL_UNIT? to ?QO_UNIT?, Temp  
T_Oil_Test, Pres P_Oil_Test
```

The next stage involves a trick to scale the incoming molar streams by a value equal to the ratio of the measured volume to the volume calculated for this molar stream. In other words we are honoring the measured GOR. The trick is to convert the molar stream into a characterization that has 2 extra components than the original EOS characterization (EOS+V). The 1<sup>st</sup> extra component is the *negative* sum of all EOS components. The 2<sup>nd</sup> extra component is the calculated volume for the molar stream.

Describing the procedure for the vapor from SP1, the Stage QG1 uses the separator QG1 that multiplies all incoming streams by the value of the

associated variable `Qg_Data` (measured Gas rate). This multiplication is accomplished by the following definition of the separator:

```
SEPARATOR QG1 SPLITS
SET Qg_Data 0 ?QG_UNIT?
LIQUID ?NC?*0, VAPOR ?NC?*0
SET Qg_Data 1.E30 ?QG_UNIT?
LIQUID ?NC?*1.E30, VAPOR ?NC?*0
```

The feed to QG1 comes from vapor of SP1 (making up EOS components) and all from SG1(making up the 2<sup>nd</sup> extra volume component). The conversion also specifies that -1 times all the component quantities should end up in the 1<sup>st</sup> extra component (named OFFSET). During the Process the feeds are **normalized**. This means that the feed is divided by the sum of the component. Since individual components and the component OFFSET offset each other, the total effect is division by the calculated gas volume. Coupled with the multiplication by the measured gas volume, we achieve the effect of scaling the stream by a ratio of  $Q_g^M / Q_g^C$ . But we still have the extra 2 components to get rid off. This is accomplished by specifying that these components do not get carried over to the normal EOS characterization upon conversion when feeding to the stage SP2.

A similar strategy is used for oil. An additional stage QO2 (using separator QO2) multiplies the resulting molar stream from QO1 by the variable used for the reported “shrinkage factor”.

The “corrected” moles of oil and gas are fed together into the “reference” separator Sep2 at stage SP2. The split vapor and liquid are processed through stages SG2 and SO2 to get gas and oil volumes respectively. The additional stage ST2 converts the separator oil to stock-tank conditions (this time for the reference separator) after which stage TO2 converts it to a single component oil volume characterization using the reducer:

```
REDUCE "?CHAR_NAME?" from ?MOL_UNIT? to ?QO_UNIT?, Temp
T_Oil_Ref, Pres P_Oil_Ref
SEP TOIL2, Splits
```

The vapor from SP2 is converted to surface gas volumes in stage SG2. A scale/normalize trick is used to calculate the GOR. The conversion from stage SG2 (characterization SGAS) specifies a *negative* SGAS into the stage SGR (characterization SGOR):

```
CONVERT SGAS FROM VOLUME TO AMOUNT
SPLIT SGAS SGAS 1 1e-15 -1
```

The feed to SGR also comes from stage SO2 (volume of oil). When normalized, this stage results in each component of this stage (i.e. SGAS, SOIL2 and -SGAS) being divided by the total (i.e. SOIL2). Subsequently,

when the conversion from this stage to the “SepTest” characterization is defined, it uses only the first component (SGAS, i.e. the volume of gas) which has already been normalized (i.e. divided by SOIL2).

```
CONVERT SGOR FROM AMOUNT TO AMOUNT, WARNINGS OFF
SPLIT SGAS GOR_sep
```

Hence the correct GOR of the separator is calculated and used in the final characterization for the component “GOR\_sep”.

A similar strategy is used for the total GOR where the additional gas, liberated from taking the separator oil to stock-tank conditions, is added to the separator gas. The same trick gives the “GOR\_tot” component in the final characterization.

The next stage TML basically collects the vapor and liquid from the stage SP2. This stage is a **splits** separator TML that passes everything and is linked to the characterization EOS+V. The conversion from characterization EOS (stage SP2) to EOS+V basically reproduces the individual components, but also creates a component OFFSET which is the *negative* sum of all the components:

```
CONVERT "?CHAR_NAME?" WARNINGS OFF
SPLIT N2      N2      OFFSET -1
.
.
.
SPLIT C7+3   C7+3   OFFSET -1
```

When this stage feeds to the final output file with the characterization “SepTest”, the conversion specifies a *negative* amount of this negative component OFFSET:

```
CONVERT EOS+V FROM MOLES TO AMOUNT, WARNINGS OFF
SPLIT OFFSET ?MOL_UNIT? -1
```

The result is that the total moles of all the components are carried by the component names ?MOL\_UNIT? (e.g. LBMOL) in the characterization.

The last components of the “SepTest” characterization are the original EOS components. Since we started with compositions, we choose to write out the compositions. This is accomplished by the stage NRM, which is a **splits** type separator “PASS” with a characterization EOS. It takes **all** (vapor and liquid) feed from stage SP2 which is also of characterization EOS. So no conversions are needed. The only action occurring at this stage is a **normalization**. This results in molar compositions corresponding to the **corrected** feed at stage SP2. This is written to the output file as the last part of the characterization “SepTest”.

### 3.4.10 Conclusions

The Streamz program, the only software of its kind providing easy, accurate, consistent and efficient conversion, management and processing of petroleum fluid streams, has been explained in detail. Its usage in the examples used in this thesis has been dissected line by line. Specific commands are discussed in the context of their purpose and utility.

The ability of Streamz to solve any petroleum fluid stream related problem has been confirmed by its use on a variety of stream conversions. Its flexible and generic nature allows its usage on problems it was not specifically programmed to handle. Its ability to handle huge number of streams resulting from multiple conversions of multiple reservoir models, converting, filtering, aggregating and processing as required, has demonstrated its robustness.



## References

1. Fevang, Ø, Singh, K, and Whitson, C. H. SPE 63087 *Guidelines for choosing Compositional and Black-Oil Models for Volatile Oil and Gas-Condensate Reservoirs*, SPE Annual Technical Conference and Exhibition, Dallas, October, 2000.
2. Coats, Keith H., SPE10512 *Simulation of Gas Condensate Reservoir Performance*, 1985.
3. Leibovici, C. F., Barker, J. W. and Wachě, D, *Method of Delumping the Results of Compositional Reservoir Simulation*, SPE64001.
4. Schlijper, A. G., Drohm, J. K., *Inverse Lumping: Estimating Compositional Data from Lumped Information*, SPE14267, SPE Reservoir Engineering, August 1998.
5. Whitson, C. H., Anderson, T F., and Søriede, I., *C<sub>7+</sub> Characterization of Related Equilibrium Fluids Using the Gamma Distribution*, Advances in Thermodynamics Volume 1, C<sub>7+</sub> Fraction Characterization, L.G. Chorn and G.A. Mansoori (ed.), Advances in Thermodynamics, Taylor & Francis, New York, ISBN 0-8448-1565-9.
6. Whitson, C. H. and Brule, M. R.: Phase Behavior. SPE Monograph Volume 20, Henry L. Doherty Series, 2000.
7. Søriede, I, PhD Thesis, *Improved Phase Behavior Predictions of Petroleum Reservoir Fluids from a Cubic Equation of State*, April 1989.
8. Zick, A. A., Personal discussions.
9. Zick, A. A., personal discussions and demonstration during development of PhazeComp, a unique PVT program allowing a variety of equation of state calculations including MMP calculations.
10. Zick, A. A., *PhazeComp: Zick Technologies' new program for compositional phase behavior computations using an equation of state (EOS)*, Product description brochure.
11. Whitson, C.H. and Torp, S.B.: "Evaluating Constant Volume Depletion Data", *JPT* (March 1983) 610; *Trans., AIME*, **275**.
12. Whitson, C.H. and Søriede, I., *Simulating Compositionally Sensitive Reservoir Processes (Preliminary Report)*, March 1989.
13. Rovere, A., Sabathier, J.C. and Codreanu, D.B., *Modification of a Black Oil Model for Simulation of Volatile Oil Reservoirs*, Third European Meeting on Improved Oil Recovery, 1983.
14. Zick, Aaron A, In-house presentation April 2001.

15. Coats Engineering, Inc., *Sensor Manual*, Jan 2001.
16. Hoda, M. F., *File Format for Extended Black-Oil PVT Data*, June 2001.
17. Uleberg, K., *Adaptation of Petrostream Management BO<sub>z</sub> method to CO<sub>2</sub>, CO<sub>2</sub>-swelled oil and reservoir oil*, internal circulation.
18. Hoda, M. Faizul, *PetroStream Management - Getting Started with Stream<sub>z</sub> Version 1.0.*, Jan 2002.
19. Hoda, M. F., *PetroStream Management - User Manual Stream<sub>z</sub> Version 1.0*, Jan 2002.
20. Hoda, M. F., *PetroStream Management - Getting Started with PSM Toolkit*, June 2001.
21. Whitson C.H., *PVT<sub>x</sub> : An Equation-of-State Based Program for Simulating & Matching PVT Experiments with Multiparameter Nonlinear Regression*, version 2000.
22. Eclipse 100, 1998-2001 Releases.
23. Eclipse 300, 1998-2001 Releases.

## Nomenclature

16c	=	16 component
2D	=	Two dimensional
3D	=	Three dimensional
9c	=	9 component
$\infty$	=	Infinity
A	=	Function of OGR
A'	=	Derivative of A
B	=	Function of OGR
B'	=	Derivative of B
B <sub>g</sub>	=	Gas formation volume factor
B <sub>o</sub>	=	Oil formation volume factor
BO	=	Black Oil
BOz	=	Black-oil to compositional
Boz2Cnv	=	PSM module that automatically generates split factors for conversion from black-oil to molar rates.
C <sub>7+</sub>	=	Heptane and heavier components
CCE	=	Constant Composition Expansion experiment
CGFR	=	Eclipse keyword denoting connection gas flow rate
CME	=	Constant Mass Expansion experiment
CNV	=	Conversion (Split factor) tables or files
COFR	=	Eclipse keyword denoting connection oil flow rate
C <sub>og</sub>	=	Conversion of surface oil from reservoir gas volume to an equivalent surface gas volume
C <sub>oo</sub>	=	Conversion of surface oil from reservoir oil volume to an equivalent surface gas volume
CVD	=	Constant Volume Depletion experiment
D	=	Day(s)
DLE	=	Differential Liberation experiment

---

dM	=	The derivative of the Molecular weight variable
e	=	Exponentiation
ECL	=	Reservoir simulator Eclipse
EG	=	Equilibrium Gas
EO	=	Equilibrium Oil
EOS	=	Equation of State
exp	=	Exponentiation
f	=	Function
f'	=	Derivative of the function f
FGPR	=	Eclipse keyword denoting field gas production rate
F <sub>oo</sub>	=	Volume fraction of surface oil that comes from reservoir oil
FOPR	=	Eclipse keyword denoting field oil production rate
g	=	Function
g'	=	Derivative of the function g
GC	=	Gas Chromatography
	=	Gas Condensate
GD	=	Gamma Distribution
GOR	=	Gas oil ratio
IG	=	Injection Gas
j	=	Counter for summation of the infinite series representation of the integral of p(M)
k	=	Counter for product in the infinite series representation of the integral of p(M)
	=	Unit dependent Constant in the split factor formulation of BOz method
K	=	Necessarily positive constant
K <sub>i</sub>	=	Equilibrium K-value for component i
lb-mol	=	Pound moles
LMW	=	Lower molecular weight
M	=	Molecular weight
M <sub>7+</sub>	=	Average molecular weight of the C <sub>7+</sub> components

$M_b$	=	Bounding molecular weight
$M_{bi}$	=	Bounding molecular weight of the $i^{\text{th}}$ fraction
$M_{bi-1}$	=	Bounding molecular weight of the $(i-1)^{\text{th}}$ fraction
$m_i$	=	Moles of $i^{\text{th}}$ component in wellstream
$M_o$	=	Molecular weight of oil
Mscf	=	Thousand Standard Cubic Feet
MW	=	Molecular weight
$n_g$	=	Total moles of gas
$n_{gg}$	=	Moles of surface gas from reservoir gas
$n_{gi}$	=	Moles of $i^{\text{th}}$ component in gas
$n_{go}$	=	Moles of surface gas from reservoir oil
$n_i$	=	Moles of $i^{\text{th}}$ component in wellstream
$n_o$	=	Total moles of oil
$n_{og}$	=	Moles of surface oil from reservoir gas
$n_{oi}$	=	Moles of $i^{\text{th}}$ component in oil
$n_{oo}$	=	Moles of surface oil from reservoir oil
$OGR_{BO}$	=	Oil Gas ratio of a black oil stream (e.g. from a BO simulator)
OGRP	=	Producing Oil Gas ratio
$OGR_{PSM}$	=	Oil Gas ratio of a BOz stream processed through a surface process, resulting in oil and gas.
$p(M)$	=	Probability density function of M
$P_0(M)$	=	The cumulative probability density function i.e. the integral of $p(M)$
$P_1(M)$	=	Integral of the M times the Gamma distribution (probability density) function
PSM	=	Petroleum Stream Management software
PVT	=	Pressure Volume Temperature
Q	=	The fixed portion of the infinite series representation of the integral of $p(M)$
$q_g$	=	Total surface gas Volume (rate)
$q_{gg}$	=	Volume (rate) of surface gas from reservoir gas

---

$q_{go}$	=	Volume (rate) of surface gas from reservoir oil
$q_j$	=	Quantity of input component (either gas or oil) in the BOz method
$q_o$	=	Total surface oil
$q_{og}$	=	Volume (rate) of surface oil from reservoir gas
$q_{oo}$	=	Volume (rate) of surface oil from reservoir oil
$q_{sg}$	=	Surface volumetric gas rate
$q_{so}$	=	Surface volumetric oil rate
RG	=	Reservoir Gas
RO	=	Reservoir Oil
$R_p$	=	Producing Gas-Oil ratio
$R_s$	=	Solution Gas-Oil ratio
$r_s$	=	Solution Oil-Gas ratio
$R_v$	=	Solution Oil-Gas ratio
S	=	The “sum over infinite” portion of infinite series representation of the integral of $p(M)$
SCN	=	Single Carbon Number
SEP	=	Separator experiment
SG	=	Surface gas
$S_{ig}$	=	Split factor for the gas input component to split into the $i^{\text{th}}$ output molar component
$S_{ij}$	=	Split factor of the $i^{\text{th}}$ input component that splits into the $j^{\text{th}}$ output component
$S_{io}$	=	Split factor for the oil input component to split into the $i^{\text{th}}$ output molar component
SO	=	Surface oil
STB	=	Stock Tank Barrel
TBP	=	True Boiling Point
$u_i$	=	Amount of output component $i$
$V_g$	=	Reservoir gas volume
$v_j$	=	Amount of input component $j$
VLE	=	Vapor Liquid Equilibria experiment

$V_o$	=	Reservoir oil volume
WBHP	=	Eclipse keyword denoting well bottom hole pressure
WGPR	=	Eclipse keyword denoting well gas production rate
WOPR	=	Eclipse keyword denoting well oil production rate
$x_i$	=	Mole fraction of $i^{\text{th}}$ component in the oil phase
$y$	=	Composite parameter defined as $(M_b - \eta) / \beta$ . Used in the infinite series representation of the integral of $p(M)$
$y_g$	=	Surface gas component
$y_i$	=	Mole fraction of $i^{\text{th}}$ component in the gas phase
$y_o$	=	Surface oil component
$z$	=	Mole fraction of wellstream
$z_{7+}$	=	Total mole fraction of the $C_{7+}$ components
$z_i$	=	Mole fraction of the $i^{\text{th}}$ fraction or component
$z_i / z_{7+}$	=	Normalized mole fraction of the $i^{\text{th}}$ fraction
$\alpha$	=	Alpha; The shape parameter of the Gamma Distribution
$\beta$	=	Beta; A composite parameter of the Gamma Distribution
	=	Fraction of wellstream coming from gas phase
$\beta_1$	=	Vapor mole fraction of the 1 <sup>st</sup> stage separator
$\beta_2$	=	Vapor mole fraction of the 2 <sup>nd</sup> stage separator
$\beta_3$	=	Vapor mole fraction of the 3 <sup>rd</sup> stage separator
$\beta_g$	=	Mole fraction of surface gas in reservoir gas
$\beta_o$	=	Mole fraction of surface gas in reservoir oil
$\Gamma$	=	Gamma function
$\gamma_{7+}$	=	Average Specific Gravity of $C_{7+}$ components
$\eta$	=	Eta; Value of molecular weight at which $p(M)$ is zero
$\rho_o$	=	Density of oil



## Appendix A

### Typical Test and Reference Separator Ratios

This appendix develops the relations for conversions between a typical test separator system and a reference separator system. This relation is used in the correction of well test data to a common basis.



**Ratios between a typical Test separator and a reference separator**

When a reservoir phase (either gas or oil) is flashed through a surface separator (e.g. a Test Separator) system the fraction of surface oil,  $F_o$ , is given by (ref. Whitson & Brule):

$$F_o = \frac{n_o}{n} = \left( 1 + \frac{GOR}{C_o} \right)^{-1} \dots\dots\dots (1)$$

Here GOR equals  $R_s$  if the reservoir phase is oil and equals  $1/R_v$  if the reservoir phase is gas.  $C_o$  is the conversion from surface oil volume to an equivalent surface gas volume and equals 132883.29907 scf/STB in field units. The corresponding fraction of gas is:

$$F_g = \frac{n_g}{n} = 1 - F_o \dots\dots\dots (2)$$

An expression for the volume of oil can be written as:

$$q_o = \frac{m_o}{\rho_o} = \frac{m_o}{M_o} \frac{M_o}{\gamma_o \rho_w} = n_o \frac{M_o}{\gamma_o \rho_w} \dots\dots\dots (3)$$

$\rho_w$  is the standard water density. Considering 1 mole ( $n_o=F_o$ ) of the phase,

$$q_o = F_o \frac{M_o}{\gamma_o \rho_w} \dots\dots\dots (4)$$

For the volume of gas the expression is simply:

$$q_g = (1 - F_o)k \dots\dots\dots (5)$$

Where the constant **k** is the constant gas volume occupied by a mole of any gas at standard temperature and pressure. It has the familiar values of 23.69024531 (m3/kmol) in metric units and 379.4813268 (ft3/lbmol) in field units.

The above equations for  $q_o$  and  $q_g$  are valid for any separator system. If the separator system were a hypothetical single stage separator, the values of volumes of oil and gas would be:

$$q_{o,ref} = F_{o,ref} \left( \frac{M_o}{\gamma_o} \right)_{ref} \frac{1}{\rho_w} \dots\dots\dots (6)$$

Similar equation for gas becomes

$$q_{g,ref} = (1 - F_{o,ref})k \dots\dots\dots (7)$$

A ratio of “oil volume processed through any Test Separator system” and “oil volume processed through a Single Stage Separator system” would be

$$f_o = \frac{q_o}{q_{o,ref}} = \frac{F_o (M_o / \gamma_o)}{F_{o,ref} (M_o / \gamma_o)_{ref}} \dots\dots\dots (8)$$

Similarly for gas,

$$f_g = \frac{q_g}{q_{g,ref}} = \frac{1 - F_o}{1 - F_{o,ref}} \dots\dots\dots (9)$$

Considering a conversion from surface oil volume through a Test Separator ( $q_o$ ) to a surface oil volume through a Single Stage separator ( $q_{o,ref}$ ), and similarly for gas, Eqs. (8) and (9) can be written in the format of split factors:

$$q_{o,ref} = \frac{1}{f_o} q_o + 0 \cdot q_g \dots\dots\dots (10)$$

$$q_{g,ref} = 0 \cdot q_o + \frac{1}{f_g} q_g \dots\dots\dots (11)$$

When converting from the Single Stage separator to the common conditions Test Separator the equations become:

$$q_o = f_o q_{o,ref} + 0 \cdot q_{g,ref} \dots\dots\dots (12)$$

$$q_g = 0 \cdot q_o + f_g q_{g,ref} \dots\dots\dots (13)$$

The physical properties required for the conversions vary with the Test Separator conditions. For each condition ( $p_{sp}$ ,  $T_{sp}$ ), an EOS-based PVT program needs to calculate the terms  $f_o$  and  $f_g$ , which depend on  $F_o$ ,  $F_{o,ref}$ ,

$M_o/\gamma_o$ , and  $(M_o/\gamma_o)_{ref}$ . These can be calculated by the EOS-based PVT program for any condition of  $p_{sp}$  and  $T_{sp}$ . For implementation with a pre-calculated set of split factors, these should be calculated for a range of  $p$  and  $T$  space that are expected to be used for the Test separator. When used for any  $p$  and  $T$ , the actual values of these parameters will be interpolated.



## **Appendix B**

### **File Format for Extended Black-Oil PVT Data**

This appendix contains the published format for reporting of extended black-oil PVT data from a EOS-based PVT program. This data is then used by the BOZ conversions method for conversion of surface black-oil rates to compositional molar rates.

This is a standard and circulated document. As such is being included in its original format and does not comply to the format of the rest of this thesis.



### **File format for extended black-oil PVT data**

This document describes the file format for exchange of data between a PVT program and the PetroStream Management software.

**Purpose:** File format for data needed by PetroStream Management software to create tables for conversion of black-oil (BO) to compositional streams. Recommended extension of the file is “**boz**”.

**Source:** An EOS based PVT model or software that generates black-oil PVT tables.

**Description:** The **.boz** file created using the format described in this note is generated at the same time that an EOS-based PVT program generates a black-oil PVT table.

When the black-oil PVT table is generated a “depletion-type” experiment is simulated – e.g. constant composition expansion, differential liberation test, or constant volume depletion test. The black-oil PVT table contains a number of saturated PVT data and, usually, some undersaturated data.

The “extended” black-oil information written to the **.boz** file uses *only* information from the saturated conditions during the depletion experiment. In particular, the data needed for each “saturated” pressure are:

- (1) solution oil-gas ratio,  $r_s$  or  $R_v$ , for the reservoir gas phase;
- (2) molar density factor,  $C_{og}$ , of the reservoir gas phase, for converting surface oil volume to an “equivalent” surface gas volume;
- (3) the equilibrium gas composition of the reservoir gas phase;
  
- (4) inverse solution gas-oil ratio,  $1/R_s$ , for the reservoir oil phase;
- (5) molar density factor,  $C_{oo}$ , of the reservoir oil phase, for converting surface oil volume to an “equivalent” surface gas volume;
- (6) the equilibrium oil composition of the reservoir oil phase.

Content:

**A. Header Section**

Occurs fully before the "Extended black-oil PVT data" section (see the example), and consists of the reserved keywords:

- i) **TITLE** (optional), followed by a string within single quotes, typically identifying the source of the file.
- ii) **CHAR** (required), followed by a characterization name. If it contains embedded spaces, the name should be within single quotes.
- iii) **EOS** (required), followed by one of the recognized names identifying the equation of state. Recognized EOS names are:

**RK** Original Redlich-Kwong EOS  
**SRK** Soave-Redlich-Kwong EOS  
**PR77** Original Peng-Robinson EOS  
**PR** Modified Peng-Robinson EOS (default)

- iv) The full property table (required) for the EOS characterization, consisting of:

a) A row of headings of the following properties:

**COMP** (component name; max. 8 alphanumeric characters)  
**MW** (molecular weight)  
**TC** (critical temperature)  
**PC** (critical pressure)  
**AF** (acentric factor)  
**VS** (volume shift or translation factor)  
**AMOD** (A-parameter multiplier)  
**BMOD** (B-parameter multiplier)

b) A row of units for **PC** and **TC**. The units allowed for **TC** are **R**, **K**, **F**, **C**, and for **PC** are **ATM**, **PSI**, **BAR**, **KPA**, **MPA**, **TORR** (absolute pressures are indicated by these keywords as shown, or with an added **A** - e.g. **PSIA**; gauge pressures are indicated by an added **G** - e.g. **PSIG**). None of the other properties have units associated with them. The units should *line up* with their respective headings. The meaning of *line up* is described at the end of this document.

c) Rows of actual property data. These should *line up* with their respective headings. The meaning of *line up* is described at the end of this document. A blank line or the keywords **BIPS** or **END** completes a property table.

- d) Binary interaction parameter tables (required), initiated using the **BIPS** keyword followed by a similarly formatted table. The headings following the **BIPS** keyword are the names of the components specified in the property table. The first column of each subsequent row also contains the names of components. Corresponding **BIP** data are entered at the intersection of the respective column and row headings. These data should also *line up*. A blank line or the word **END** completes a binary interaction parameter table.
- v) Optional "comments" using the semi-colon (;). Any characters to the right of a semi-colon will be ignored.

#### **B. Extended black-oil PVT data Section**

Starts after the "Header" section ends, and consists of:

- i) The keyword **UNIT**, followed by either **FIELD** or **METRIC**, specifying the units in which the extended black-oil PVT data has been printed.
- ii) The keyword **ID**, followed by a contiguous (no blanks) string or "token" uniquely identifying the table that follows this identifier. This "token" should be not more than 20 characters long; the first table in the example uses **CVD1\_200\_PVTG**. The token may be as simple as a unique integer associated with the table. This may be followed on the same line by an optional string, within single quotes, typically giving a description of the source and contents of the identified table. The length can be 132 characters. The first table in the example uses: '**CVD Experiment 1, Feed: 200, Gas Phase**' as a descriptor.
- iii) Optional lines of comments using semi-colons, to provide a table heading for the identified table. This is ignored completely by the program but may provide important information to the user (e.g. units used).
- iv) A table of extended PVT data containing, in order, the depletion pressure, solution oil-gas ratio, the molar density, and the phase molar compositions (all defined later). One table associated with a unique ID should be generated by the PVT program for each phase in each experiment and consist of extended PVT data at all the saturated pressure nodes specified in the experiment. All data for a particular pressure should be on a single line. One table consists of multiple lines of such data for all the pressures associated with this ID.

Definitions: The following terms are used in the description above:

- a) **Pressure** for a given stage of depletion in the experiment (only saturated values should be used).

- b) **Oil-gas ratio** is the volumetric ratio of surface oil to surface gas, produced from a reservoir gas phase ( $r_s$  or  $R_v$ ) or a reservoir oil phase ( $1/R_s$ ), at the specified pressure. If the phase is an oil, the *inverse* of the traditional solution gas-oil ratio should be printed.
- c) **Molar density factor** is a conversion from a surface oil volume to an "equivalent" surface gas volume, defined as:

$$C_{\bar{o}x} = k \cdot \frac{\gamma_{\bar{o}x}}{M_{\bar{o}x}}$$

where:

- $\bar{o}$  : Subscript denoting surface oil
- x : Subscript denoting either o (for reservoir oil) or g (for reservoir gas).
- $\gamma$  : Specific gravity of surface oil from the relevant phase
- M : Molecular weight of surface oil from the relevant phase.
- k : **132883.29907** in **FIELD** units (psia, STB/Mscf, and scf/STB).  
**23667.52637** in **METRIC** units (bara, Sm<sup>3</sup>/Sm<sup>3</sup>, and Sm<sup>3</sup>/Sm<sup>3</sup>)<sup>1</sup>

- d) **Phase compositions** are *mole fractions* of the components of the phase at the depletion pressure.

Data format: The numeric data on the file can be separated with spaces or commas, and supports normal Fortran input format. All input shown in examples on one line, should *not* extend to multiple lines. The lines are read-in as strings, so there are no field width limitations. Any numeric data, separated by delimiters listed above, can use the width required by the precision. Internally these numeric data are converted to double precision. We suggest using E format with six significant digits.

Example: Example of the file containing the extended black-oil table, adapted from an output from the in-house Pera-PVTx program.

---

<sup>1</sup> The precision provided above for  $k=k_1(\rho_w RT_{sc}/p_{sc})$  may be important for near-critical systems. For field units,  $k$  is based on the following constants:

Conversion factor for oil volume to gas volume in field units,  $k_1 = 5.614583$  (ft<sup>3</sup>/bbl).  
 Water Density at Standard conditions,  $\rho_w = 0.999041$  (g/cc).  
 Universal gas constant,  $R = 10.7315$  (psi-ft<sup>3</sup>/lbmole-°R).  
 Standard Temperature,  $T_{sc} = 520^\circ\text{R}$  and Pressure,  $p_{sc} = 14.69595$  (psia).

a) Header section:

TITLE 'Extended black-oil PVT data generated by PVTx'

CHAR 'EOSname'

EOS SRK

COMP	MW	TC R	PC PSIA	AF	VS	AMOD	BMOD
N2	28.01	227.27	493.00	0.0500	-0.1930	1.0000	1.0000
CO2	44.01	547.57	1070.60	0.2300	-0.0820	1.0000	1.0000
C1	16.04	343.04	667.80	0.0100	-0.1590	1.0000	1.0000
C2	30.07	549.76	707.80	0.0900	-0.1130	1.0000	1.0000
C3	44.10	665.68	616.30	0.1500	-0.0860	1.0000	1.0000
IC4	58.12	734.65	529.10	0.1800	-0.0840	1.0000	1.0000
C4	58.12	765.32	550.70	0.1900	-0.0670	1.0000	1.0000
IC5	72.15	828.77	490.40	0.2300	-0.0610	1.0000	1.0000
C5	72.15	845.37	488.60	0.2500	-0.0390	1.0000	1.0000
C6	86.18	913.37	436.90	0.3000	-0.0080	1.0000	1.0000
F1	98.55	1002.61	437.36	0.2900	0.0353	1.0000	1.0000
F2	135.84	1132.41	359.15	0.3900	0.0581	1.0000	1.0000
F3	206.65	1306.40	263.24	0.5800	0.1140	1.0000	1.0000

BIPS	NO2	CO2	C1	C2	C3	IC4	C4	IC5	C5	C6	F1	F2	F3
N2	0.0000	0.0000	0.0250	0.0100	0.0900	0.0950	0.0950	0.1000	0.1100	0.1100	0.1100	0.1100	0.1100
CO2	0.0000	0.0000	0.1050	0.1300	0.1250	0.1200	0.1150	0.1150	0.1150	0.1150	0.1150	0.1150	0.1150
C1	0.0250	0.1050	0.0000	0.0826	-0.1120	0.0265	0.0000	0.0000	0.0000	0.0000	0.0014	0.0019	0.0027
C2	0.0100	0.1300	0.0826	0.0000	0.0130	0.0170	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
C3	0.0900	0.1250	-0.1120	0.0130	0.0000	0.0200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
IC4	0.0950	0.1200	0.0265	0.0170	0.0200	0.0000	0.0149	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
C4	0.0950	0.1150	0.0000	0.0000	0.0000	0.0149	0.0000	0.0129	0.0000	0.0000	0.0000	0.0000	0.0000
IC5	0.1000	0.1150	0.0000	0.0000	0.0000	0.0000	0.0129	0.0000	0.0000	0.0120	0.0000	0.0000	0.0000
C5	0.1100	0.1150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
C6	0.1100	0.1150	0.0000	0.0000	0.0000	0.0000	0.0000	0.0120	0.0000	0.0000	0.0037	0.0000	0.0000
F1	0.1100	0.1150	0.0014	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0037	0.0000	-0.1280	-0.2000
F2	0.1100	0.1150	0.0019	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.1280	0.0000	0.0000
F3	0.1100	0.1150	0.0027	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.2000	0.0000	0.0000



## b) Extended Black-Oil Data section (comments are in Arial font, red color):

```

UNIT FIELD
ID CVD1_200_PVTG 'CVD Experiment 1, Feed: 200, Gas Phase'
;-----
;- Phase      Oil-Gas      Molar
;- Pressure   Ratio          Density   Compositions ---(13 entries for this example with 13 components, all on single line) --->
;- (psia)    (STB/Mscf)    (scf/STB) (SI units: bara, Sm3/Sm3, Sm3/Sm3)
;-----
3415.692  0.0436577  0.721687E+03 .131745E-01 .852545E-01 .656477E+00 .869266E-01 .695459E-01 .149018E-01 .193457E-01 .629700E-02 ...
2500.000  0.0319218  0.748238E+03 .133285E-01 .858439E-01 .662250E+00 .872151E-01 .694867E-01 .148060E-01 .191783E-01 .620171E-02 ...
1000.000  0.0178626  0.789971E+03 .133927E-01 .865728E-01 .667040E+00 .881149E-01 .701603E-01 .149469E-01 .193431E-01 .622327E-02 ...
500.000   0.0208771  0.798085E+03 .132589E-01 .862112E-01 .662606E+00 .882005E-01 .707836E-01 .152227E-01 .197836E-01 .644602E-02 ...

ID CVD1_200_PVTO 'CVD Experiment 1, Feed: 200, Oil Phase'
;-----
;- Phase      Oil-Gas      Molar
;- Pressure   Ratio          Density   Compositions---->
;- (psia)    (STB/Mscf)    (scf/STB)
;-----
3415.692  0.8776122  0.582228E+03 .551227E-02 .593415E-01 .388629E+00 .769161E-01 .723952E-01 .193928E-01 .268878E-01 .102935E-01 ...
2500.000  1.1449839  0.637073E+03 .413832E-02 .507889E-01 .318340E+00 .701567E-01 .730601E-01 .205228E-01 .291569E-01 .118711E-01 ...
1000.000  3.6599915  0.647743E+03 .146857E-02 .245533E-01 .138471E+00 .397062E-01 .523458E-01 .166829E-01 .250642E-01 .118771E-01 ...
500.000   8.7039873  0.631279E+03 .695478E-03 .129880E-01 .704445E-01 .223731E-01 .326014E-01 .110059E-01 .169360E-01 .864514E-02 ...

ID DLE1_100_PVTG 'DLE Experiment 1, Feed: 100, Gas Phase'
;-----
;- Phase      Oil-Gas      Molar
;- Pressure   Ratio          Density   Compositions---->
;- (psia)    (STB/Mscf)    (scf/STB)
;-----
3415.658  0.0436574  0.721687E+03 .131744E-01 .852546E-01 .656477E+00 .869266E-01 .695458E-01 .149018E-01 .193457E-01 .629702E-02 ...
2500.000  0.0254173  0.748219E+03 .115436E-01 .883286E-01 .656622E+00 .929616E-01 .717792E-01 .154049E-01 .197698E-01 .622291E-02 ...
1000.000  0.0132042  0.791247E+03 .592606E-02 .957650E-01 .592029E+00 .120986E+00 .946640E-01 .211432E-01 .268718E-01 .801350E-02 ...
500.000   0.0168907  0.815168E+03 .325829E-02 .952573E-01 .497836E+00 .145760E+00 .128206E+00 .305766E-01 .394050E-01 .118683E-01 ...

ID DLE1_100_PVTO 'DLE Experiment 1, Feed: 100, Oil Phase'
3415.658  0.8776046  0.582233E+03 .551220E-02 .593415E-01 .388629E+00 .769161E-01 .723954E-01 .193928E-01 .268879E-01 .102936E-01 ...
2500.000  1.3618690  0.579366E+03 .334715E-02 .500596E-01 .299791E+00 .722414E-01 .728668E-01 .207729E-01 .292999E-01 .116381E-01 ...
1000.000  3.5415569  0.576299E+03 .645140E-03 .266216E-01 .120902E+00 .527661E-01 .674243E-01 .222166E-01 .327050E-01 .141799E-01 ...
500.000   6.4405864  0.574965E+03 .173633E-03 .142373E-01 .528891E-01 .359865E-01 .564571E-01 .207081E-01 .314961E-01 .145970E-01 ...

```



Notes for EOS Property and BIPs tables – How to Line Up data columns

The EOS property table consists of a row of headings, a row of units, and rows of component values. A component value is associated with a heading according to its “line up”.

**Line Up Procedure for Defining Columns**

Columns are defined by headings in the first row of a table. Subsequent lines contain table entries or “values” that are associated with a column in the following logic:

- (1) Starting at the right-most character in the value,
- (2) Moving vertically upwards to the heading row,
- (3a) If a heading is intersected, use that heading or
- (3b) If a heading is not intersected, move left on the heading row until a heading is intersected.

Units in the second row are associated with headings using the same “line up” procedure.

For the BIP table and associating components to a  $k_{ij}$  value, the component name for  $i$  is taken as the first entry in the row, and the component name for  $j$  is found using the table “line up” procedure described above.

Example of table line-up:

NAME	MW	TC	PC	AF	VS
		R	PSIA		
CO2	44.01	547.57	1071.60	0.2250	0.02700
METHANE	16.04	343.04	667.80	0.0130	-0.11800
ETHANE	30.07	549.76	707.80	0.0986	-0.10700
PROPANE	44.10	665.68	616.30	0.1524	-0.08477
N-BUTANE	58.12	765.32	550.70	0.2010	-0.06858
N-PENTAN	72.15	845.37	488.60	0.2539	-0.04103
C6	94.20	975.92	458.68	0.2695	-0.00076

BIPS	CO2	C1		
N2	0.0000	0.0250		
C02	0.0250	0.0100	.	.
C1	0.0100	0.0900		
.				
.				
.				



## Appendix C1

### North Sea Full Field

This appendix contains a portion of the complete data set for this example. A North Sea reservoir modeled with 2 black-oil PVT regions, about 50 wells, and approximately 1000 well-grid connections (well completions) was chosen as an example of a typical field application of the BOz method. Further complications were the existence of gas injection and 2 different surface processes. An estimated 100,000 streams were expected. The conversion was performed on a grid cell connection basis at each time step before averaging & summing to yearly-field streams.

The complete driver file is included in this appendix. To conserve space and still include a complete ready-to-run data set, only a portion of the conversion file is listed. This portion corresponds to the portion of the input stream file included in this appendix.

The complete data set is included in the accompanying CD-Rom.

**Streamz Driver File: e100-bo-9c.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:

INCLUDE 'bo.chr'

;Open the input stream file containing streams with input
;characterization:

DEFINE inp_str_file = 'e100-bo.str'

STREAMFILE INP1: INPUT ?inp_str_file?

;Include file containing the output characterization and the
;conversion definitions:

INCLUDE 'bo-9c-inj.cnv'

;Open the output stream file to contain streams with output
;characterization:

DEFINE out_str_file = 'e100-bo-9c.str'

STREAMFILE OUT1: OUTPUT ?out_str_file?

END

;Include file containing other manipulations:

INCLUDE 'tabulate.inc'
```

**Characterization Include File: bo.chr**

```
CHAR 'BO_Char'
NAME
SO
SG
```

**Characterization Include File: eos9.chr**

```
CHAR      'EOS9'
EOS        PR
COMP      MW          TC          PC          AF          VS          AMOD          BMOD

                K          BARA

C1         16.0924    190.115    45.9565    1.17E-02    -0.15913    1.00273    1.00243
CO2        44.01     304.206    73.8153    0.231       -8.20E-02    1          1
C2         30.07     305.422    48.8011    9.08E-02    -0.113      1          1
C3         44.097    369.822    42.4924    0.1454     -8.60E-02    1          1
C4         58.124    420.326    37.5454    0.187903   -7.18E-02    0.985158   1.00264
C5         72.151    465.861    33.7388    0.241262   -4.79E-02    1.00905    0.988291
C6F3       103.717    571.351    28.8451    0.315658   4.10E-02    0.966199   0.97905
F4         165.409    685.364    21.3935    0.479732   9.56E-02    1          1
F5F6       364.751    828.167    18.2002    0.710074   -0.42738    0.758841   0.927809

BIPS      C1          CO2          C2          C3          C4          C5          C6F3          F4          F5F6
C1         0.00E+00    0.104245    7.19E-05    6.47E-04    6.83E-04    7.61E-04    2.77E-02    -1.04E-02    0.179524
CO2        0.104245    0.00E+00    0.13        0.125       0.116424    0.115      0.115      0.115      0.115
C2         7.19E-05    0.13        0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00
C3         6.47E-04    0.125      0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00
C4         6.83E-04    0.116424    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00
C5         7.61E-04    0.115      0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00
C6F3       2.77E-02    0.115      0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00
F4         -1.04E-02    0.115      0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00
F5F6       0.179524    0.115      0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00    0.00E+00
```

**Conversion File: bo-9c-inj.cnv**

```

CONVERT 'BO_Char' from      VOLUMES  to      MOLES

SET      PVTNUM  1      PFLAG  2      PRES      (BARA) 280.701
SPLIT   SO      C1
        -5.09E-01 1.90E-02 2.62E-01 2.80E-01 2.32E-01
        1.92E-01 1.16E+00 1.05E+00 1.46E+00
SPLIT   SG      C1
        3.77E-02 3.75E-04 2.60E-03 1.37E-03 7.05E-04
        3.37E-04 6.48E-04 -2.41E-05 -4.36E-04

SET      PVTNUM  1      PFLAG  2      PRES      (BARA) 280.7
SPLIT   SO      C1
        -5.09E-01 1.90E-02 2.62E-01 2.80E-01 2.32E-01
        1.92E-01 1.16E+00 1.05E+00 1.46E+00
SPLIT   SG      C1
        3.77E-02 3.75E-04 2.60E-03 1.37E-03 7.05E-04
        3.37E-04 6.48E-04 -2.41E-05 -4.36E-04

SET      PVTNUM  1      PFLAG  2      PRES      (BARA) 279.74
SPLIT   SO      C1
        -5.09E-01 1.90E-02 2.62E-01 2.80E-01 2.32E-01
        1.92E-01 1.16E+00 1.05E+00 1.46E+00
SPLIT   SG      C1
        3.77E-02 3.75E-04 2.60E-03 1.37E-03 7.05E-04
        3.37E-04 6.48E-04 -2.42E-05 -4.36E-04

SET      PVTNUM  1      PFLAG  2      PRES      (BARA) 260
SPLIT   SO      C1
        -5.15E-01 1.90E-02 2.63E-01 2.81E-01 2.32E-01
        1.93E-01 1.16E+00 1.05E+00 1.46E+00
SPLIT   SG      C1
        3.77E-02 3.75E-04 2.60E-03 1.37E-03 7.05E-04
        3.37E-04 6.45E-04 -2.76E-05 -4.32E-04

SET      PVTNUM  1      PFLAG  2      PRES      (BARA) 240
SPLIT   SO      C1
        -5.20E-01 1.89E-02 2.63E-01 2.81E-01 2.32E-01
        1.93E-01 1.17E+00 1.06E+00 1.46E+00
SPLIT   SG      C1
        3.77E-02 3.75E-04 2.60E-03 1.37E-03 7.03E-04
        3.35E-04 6.36E-04 -3.80E-05 -4.20E-04

SET      PVTNUM  1      PFLAG  1      PRES      (BARA) 280.701
SPLIT   SO      C1
        1.18E+00 -6.58E-02 -1.02E+00 4.37E-01 1.34E+00
        1.01E+00 2.96E+00 9.85E-01 2.41E-01
SPLIT   SG      C1
        3.71E-02 4.05E-04 3.06E-03 1.31E-03 3.10E-04
        4.56E-05 4.22E-06 0.00E+00 0.00E+00

SET      PVTNUM  1      PFLAG  1      PRES      (BARA) 280.7
SPLIT   SO      C1
        1.18E+00 -6.58E-02 -1.02E+00 4.37E-01 1.34E+00
        1.01E+00 2.96E+00 9.85E-01 2.41E-01

```

SPLIT	SG	C1						
			3.71E-02	4.05E-04	3.06E-03	1.31E-03	3.10E-04	
			4.56E-05	4.22E-06	0.00E+00	0.00E+00		
SET	PVTNUM	1		PFLAG	1		PRES	(BARA) 279.74
SPLIT	SO	C1						
			1.18E+00	-6.60E-02	-1.02E+00	4.38E-01	1.34E+00	
			1.01E+00	2.96E+00	9.84E-01	2.38E-01		
SPLIT	SG	C1						
			3.71E-02	4.05E-04	3.06E-03	1.31E-03	3.10E-04	
			4.56E-05	4.22E-06	0.00E+00	0.00E+00		
SET	PVTNUM	1		PFLAG	1		PRES	(BARA) 260
SPLIT	SO	C1						
			1.27E+00	-7.08E-02	-1.10E+00	4.45E-01	1.40E+00	
			1.05E+00	3.06E+00	9.72E-01	1.81E-01		
SPLIT	SG	C1						
			3.71E-02	4.05E-04	3.06E-03	1.31E-03	3.10E-04	
			4.56E-05	4.22E-06	0.00E+00	0.00E+00		
SET	PVTNUM	1		PFLAG	1		PRES	(BARA) 240
SPLIT	SO	C1						
			1.35E+00	-7.70E-02	-1.19E+00	4.52E-01	1.47E+00	
			1.11E+00	3.16E+00	9.36E-01	1.31E-01		
SPLIT	SG	C1						
			3.71E-02	4.05E-04	3.06E-03	1.31E-03	3.10E-04	
			4.56E-05	4.22E-06	0.00E+00	0.00E+00		
SET	PVTNUM	2		PFLAG	2		PRES	(BARA) 280.701
SPLIT	SO	C1						
			-3.81E-01	2.00E-02	2.67E-01	2.81E-01	2.32E-01	
			1.91E-01	1.15E+00	1.04E+00	1.44E+00		
SPLIT	SG	C1						
			3.68E-02	3.69E-04	2.57E-03	1.37E-03	7.12E-04	
			3.49E-04	7.48E-04	7.95E-05	-2.84E-04		
SET	PVTNUM	2		PFLAG	2		PRES	(BARA) 280.7
SPLIT	SO	C1						
			-3.81E-01	2.00E-02	2.67E-01	2.81E-01	2.32E-01	
			1.91E-01	1.15E+00	1.04E+00	1.44E+00		
SPLIT	SG	C1						
			3.68E-02	3.69E-04	2.57E-03	1.37E-03	7.12E-04	
			3.49E-04	7.48E-04	7.95E-05	-2.84E-04		
SET	PVTNUM	2		PFLAG	2		PRES	(BARA) 279.74
SPLIT	SO	C1						
			-3.81E-01	2.00E-02	2.67E-01	2.81E-01	2.32E-01	
			1.91E-01	1.15E+00	1.04E+00	1.44E+00		
SPLIT	SG	C1						
			3.68E-02	3.69E-04	2.57E-03	1.37E-03	7.12E-04	
			3.49E-04	7.48E-04	7.93E-05	-2.83E-04		
SET	PVTNUM	2		PFLAG	2		PRES	(BARA) 260
SPLIT	SO	C1						
			-3.79E-01	2.00E-02	2.68E-01	2.82E-01	2.32E-01	

		1.91E-01	1.15E+00	1.04E+00	1.44E+00		
SPLIT	SG	C1					
		3.68E-02	3.69E-04	2.57E-03	1.36E-03	7.12E-04	
		3.48E-04	7.46E-04	7.67E-05	-2.80E-04		
SET	PVTNUM	2		PFLAG	2	PRES	(BARA) 240
SPLIT	SO	C1					
		-3.98E-01	1.98E-02	2.68E-01	2.82E-01	2.32E-01	
		1.92E-01	1.16E+00	1.04E+00	1.44E+00		
SPLIT	SG	C1					
		3.68E-02	3.69E-04	2.57E-03	1.36E-03	7.11E-04	
		3.47E-04	7.36E-04	6.56E-05	-2.70E-04		
SET	PVTNUM	2		PFLAG	1	PRES	(BARA) 280.701
SPLIT	SO	C1					
		7.18E+00	-2.66E-01	-4.02E+00	-2.33E+00	-1.94E-01	
		5.63E-01	3.63E+00	1.35E+00	3.30E-01		
SPLIT	SG	C1					
		3.49E-02	4.42E-04	3.66E-03	2.03E-03	8.21E-04	
		2.54E-04	1.14E-04	0.00E+00	0.00E+00		
SET	PVTNUM	2		PFLAG	1	PRES	(BARA) 280.7
SPLIT	SO	C1					
		7.18E+00	-2.66E-01	-4.02E+00	-2.33E+00	-1.94E-01	
		5.63E-01	3.63E+00	1.35E+00	3.30E-01		
SPLIT	SG	C1					
		3.49E-02	4.42E-04	3.66E-03	2.03E-03	8.21E-04	
		2.54E-04	1.14E-04	0.00E+00	0.00E+00		
SET	PVTNUM	2		PFLAG	1	PRES	(BARA) 279.74
SPLIT	SO	C1					
		7.21E+00	-2.67E-01	-4.03E+00	-2.34E+00	-1.96E-01	
		5.64E-01	3.64E+00	1.35E+00	3.27E-01		
SPLIT	SG	C1					
		3.49E-02	4.42E-04	3.66E-03	2.03E-03	8.21E-04	
		2.54E-04	1.14E-04	0.00E+00	0.00E+00		
SET	PVTNUM	2		PFLAG	1	PRES	(BARA) 260
SPLIT	SO	C1					
		7.79E+00	-2.90E-01	-4.37E+00	-2.54E+00	-2.30E-01	
		5.93E-01	3.83E+00	1.37E+00	2.54E-01		
SPLIT	SG	C1					
		3.49E-02	4.42E-04	3.66E-03	2.03E-03	8.21E-04	
		2.54E-04	1.14E-04	0.00E+00	0.00E+00		
SET	PVTNUM	2		PFLAG	1	PRES	(BARA) 240
SPLIT	SO	C1					
		8.56E+00	-3.19E-01	-4.82E+00	-2.82E+00	-2.79E-01	
		6.26E-01	4.04E+00	1.35E+00	1.88E-01		
SPLIT	SG	C1					
		3.49E-02	4.42E-04	3.66E-03	2.03E-03	8.21E-04	
		2.54E-04	1.14E-04	0.00E+00	0.00E+00		

**Extra Manipulation Include Files: tabulate.inc**

```
;Convert and tabulate output for on a well basis
DOMAIN TIME T1 T2
TABULATE WELL AND TIME (DAYS)

STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE

; Use tabulated well streams as new input file
STREAMFILE INP2: INPUT ?out_str_file?

; Add "-g" to group level output file extention
STREAMFILE OUT2: OUTPUT Results\grp-?out_str_file?

; Define variable FLOW as production of first component
LUMP FLOW C1

; Only include production wells
FILTER PROD FLOW MOLES GE 0
FILTER OSC_PROD FLOW MOLES GE 0 AND WELL SW 'C'
FILTER OFC_PROD FLOW MOLES GE 0 AND WELL SW 'B'
FILTER AN WELL EQ 'C-01' OR WELL EQ 'C-05' OR WELL EQ 'C-06' OR WELL
EQ 'C-06B'
OR WELL EQ 'C-09' OR WELL EQ 'C-21' OR WELL EQ 'C-08'
FILTER AN_PROD AN AND PROD

; Tabulate OSC wells
SET GROUP OSC
TABULATE GROUP AND TIME (DAYS) IF OSC_PROD

; Tabulate OFC wells
SET GROUP OFC
TABULATE GROUP AND TIME (DAYS) IF OFC_PROD

; Tabulate Alfa North wells only
SET GROUP AN
TABULATE GROUP AND TIME (DAYS) IF AN_PROD
```

**Extract of Main Stream File: ff-in.str**

```
STREAMZ          1
NOTE             'Stream file created by PSM-Toolkit'
NOTE             'Black-Oil Characterization'
NOTE             'Unit system: METRIC'
NOTE             'Calculation level: CONNECTION'
CHAR             'BO_Char'
VARIABLE FIELD   STRING
VARIABLE GROUP   STRING
VARIABLE WELL    STRING
VARIABLE PVTNUM  INTEGER
VARIABLE CONN_I  INTEGER
VARIABLE CONN_J  INTEGER
VARIABLE CONN_K  INTEGER
VARIABLE TSTEP   INTEGER
VARIABLE T1      TIME
VARIABLE T2      TIME
VARIABLE PFLAG   INTEGER
VARIABLE PRES    PRESSURE
DATA

SET              TSTEP          1
SET              T1              0 (days)
SET              T2              4916 (days)
```

WELL	CONN_I	CONN_J	CONN_K	PVTNUM	PFLAG	PRES (bara)	VOLUME SO (Sm3/day)	VOLUME SG (Sm3/day)
'C-26A'	85	39	22	1	2	243.892	34.12455	4482.176
'C-26A'	86	39	22	2	2	243.826	4.231213	549.6349
'C-26A'	86	39	21	2	2	243.625	-0.3185471	-57.00769
'C-26A'	86	38	21	2	2	243.315	23.30604	3009.784
'C-26A'	87	38	21	2	2	243.405	31.17575	4025.468
'C-26A'	87	38	20	2	2	243.304	37.40778	4833.481
'C-26A'	87	38	19	2	2	243.231	6.559815	19234.14
'C-26A'	88	38	19	2	2	243.351	181.5141	23453.78
'C-26A'	88	38	15	2	2	243.196	20.34403	80265.61
'C-26A'	89	38	15	2	2	243.438	101.9666	13228.07
'C-26A'	89	39	15	2	2	243.555	142.004	18962.46
'C-26A'	90	39	15	2	2	243.757	235.2898	30533.91
'C-26A'	90	39	14	2	2	243.573	119.1952	16452.12
'C-26A'	91	39	13	2	2	244.177	287.4223	37344.32
'C-26A'	93	39	16	2	2	245.051	128.1663	16666.55
'C-26A'	94	39	16	2	2	244.855	314.1958	40837.21
'C-26A'	95	39	15	2	2	244.75	218.5286	28396.25
'C-26A'	96	39	15	2	2	243.125	96.14233	12350.68
'C-24A'	103	39	10	2	1	0	0.00E+00	0.00E+00
'C-24A'	95	37	22	2	2	244.742	124.9196	16244.38
'C-24A'	96	37	22	2	2	244.568	37.64023	4887.974
'C-24A'	96	37	21	2	2	244.397	88.80372	18628.99



## Appendix C2

### CO<sub>2</sub> Immiscible Injection

This appendix contains a portion of the complete data set for this example. A North Sea 2D sector model containing a highly under-saturated, low API oil. The reservoir is highly permeable and contains a single horizontal well down structure. CO<sub>2</sub> is injected through an up-dip injector.

The complete driver file is included in this appendix. To conserve space and still include a complete ready-to-run data set, only a portion of the conversion file is listed. This portion corresponds to the portion of the input stream file included in this appendix.

The complete data set is included in the accompanying CD-Rom.

**Streamz Driver File: eos6-co2-bo-6c.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:

INCLUDE 'bo.chr'

;Open the input stream file containing streams with input
;characterization:

DEFINE inp_str_file = 'eos6-co2-bo.str'

STREAMFILE INP1: INPUT ?inp_str_file?

;Include file containing the output characterization and the
;conversion definitions:

INCLUDE 'spliced.cnv'

;Open the output stream file to contain streams with output
;characterization:

DEFINE out_str_file = 'Results\eos6-co2-bo-6c.str'

STREAMFILE OUT1: OUTPUT ?out_str_file?

END

;Include file containing other manipulations:

INCLUDE 'tabulate.inc'
```

**Characterization Include File: bo.chr**

```
CHAR 'BO_Char'
NAME
SO
SG
```

**Characterization Include File: eos6.chr**

```
EOS 'PR'
```

```
CHAR 'EOS_Char'
```

COMP	MW	TC K	PC BARA	AF	VS	AMOD	BMOD
C1N2	16.2292	188.851	45.7197	1.24E-2	-0.25125	1.05629	1.04401
CO2	44.01	304.206	73.8153	0.231	0.20155	1	1.00005
C2	30.07	305.422	48.8011	9.08E-2	-0.113	1	1.00005
C3C4C5	60.6959	429.562	36.303	0.20014	-6.85E-2	0.997908	0.96046
C6P1P2	134.15	641.342	26.3533	0.38266	3.99E-2	0.885228	0.994009
P3P4P5	422.243	916.339	12.618	1.0323	3.50E-2	0.88144	0.948763
END							
BIPS	C1N2	CO2	C2	C3C4C5	C6P1P2	P3P4P5	
C1N2	0.00E+0	0.10218	2.70E-4	2.63E-3	1.08E-2	1.98E-2	
CO2	0.10218	0.00E+0	0.13	0.11779	0.115	0.115	
C2	2.70E-4	0.13	0.00E+0	0.00E+0	2.33E-2	6.74E-2	
C3C4C5	2.63E-3	0.11779	0.00E+0	0.00E+0	0.00E+0	0.00E+0	
C6P1P2	1.08E-2	0.115	2.33E-2	0.00E+0	0.00E+0	0.00E+0	
P3P4P5	1.98E-2	0.115	6.74E-2	0.00E+0	0.00E+0	0.00E+0	

## Conversion File: bo-9c-inj.cnv

CHAR 'grane-eosEOS6'

EOS PR

COMP	MW	TC K	PC BARA	AF	VS	AMOD	BMOD
C1N2	16.2292	188.851	45.7197	1.24E-2	-0.25125	1.05629	1.04396
CO2	44.01	304.206	73.8153	0.231	0.20155	1	1
C2	30.07	305.422	48.8011	9.08E-2	-0.113	1	1
C3C4C5	60.6959	429.562	36.303	0.20014	-6.85E-2	0.99791	0.96041
C6P1P2	134.15	641.342	26.3533	0.38266	3.99E-2	0.88523	0.99396
P3P4P5	422.243	916.34	12.618	1.0323	3.50E-2	0.88144	0.94871

BIPS	C1N2	CO2	C2	C3C4C5	C6P1P2	P3P4P5
C1N2	0.00E+0	0.10218	2.70E-4	2.63E-3	1.08E-2	1.98E-2
CO2	0.10218	0.00E+0	0.13	0.117794	0.115	0.115
C2	2.70E-4	0.13	0.00E+0	0.00E+0	2.33E-2	6.74E-2
C3C4C5	2.63E-3	0.117794	0.00E+0	0.00E+0	0.00E+0	0.00E+0
C6P1P2	1.08E-2	0.115	2.33E-2	0.00E+0	0.00E+0	0.00E+0
P3P4P5	1.98E-2	0.115	6.74E-2	0.00E+0	0.00E+0	0.00E+0

CONVER 'BO\_Char

T ' from VOLUMES to MOLES

SET PVTNUM 1 PFLAG 2 PRES (BARA) 168.312

SPLIT	SO	C1N2
	5.47E-1	-5.78E-1
	1.92E+0	

SPLIT	SG	C1N2
	3.45E-5	4.20E-2
	1.21E-4	

SET PVTNUM 1 PFLAG 2 PRES (BARA) 53.971

SPLIT	SO	C1N2
	5.47E-1	-5.78E-1
	1.92E+0	

SPLIT	SG	C1N2
	3.45E-5	4.20E-2
	1.21E-4	

SET PVTNUM 1 PFLAG 1 PRES (BARA) 168.312

SPLIT	SO	C1N2
	5.51E-1	-6.03E-1
	1.93E+0	

SPLIT	SG	C1N2
	0.00E+0	4.22E-2
	0.00E+0	

SET PVTNUM 1 PFLAG 1 PRES (BARA) 53.971

SPLIT	SO	C1N2
	5.51E-1	-6.03E-1
	1.93E+0	

SPLIT	SG	C1N2
	0.00E+0	4.22E-2
	0.00E+0	

**Extra Manipulation Include Files: tabulate.inc**

```
domain time t1 t2
filter prod well sw 'P'
tabulate well and time days if prod
```

**Extract of Main Stream File: ff-in.str**

```
STREAMZ          1
NOTE             'Stream file created by PSM-Toolkit'
NOTE             'Black-Oil Characterization'
NOTE             'Unit system: METRIC'
NOTE             'Calculation level: CONNECTION'
CHAR             'BO_Char'
VARIABLE FIELD   STRING
VARIABLE GROUP   STRING
VARIABLE WELL    STRING
VARIABLE PVTNUM  INTEGER
VARIABLE CONN_I  INTEGER
VARIABLE CONN_J  INTEGER
VARIABLE CONN_K  INTEGER
VARIABLE TSTEP   INTEGER
VARIABLE T1      TIME
VARIABLE T2      TIME
VARIABLE PFLAG   INTEGER
VARIABLE PRES    PRESSURE
DATA

SET              TSTEP          1
SET              T1              0 (days)
```

SET T2 0.05 (days)

WELL	CONN_I	CONN_J	CONN_K	PVTNUM	PFLAG	PRES (bara)	VOLUME SO (Sm3/day)	VOLUME SG (Sm3/day)
'GINJ1'	1	14	3	1	1	171.81	0.00E+00	-351556.3
'P1'	1	8	28	1	2	53.971	2500	34585.76

SET TSTEP 2

SET T1 0.05 (days)

SET T2 50.092 (days)

WELL	CONN_I	CONN_J	CONN_K	PVTNUM	PFLAG	PRES (bara)	VOLUME SO (Sm3/day)	VOLUME SG (Sm3/day)
'GINJ1'	1	14	3	1	1	164.192	0.00E+00	-736458.5
'P1'	1	8	28	1	2	53.971	2500	34585.76

SET TSTEP 3

SET T1 50.092 (days)

SET T2 100.092 (days)

WELL	CONN_I	CONN_J	CONN_K	PVTNUM	PFLAG	PRES (bara)	VOLUME SO (Sm3/day)	VOLUME SG (Sm3/day)
'GINJ1'	1	14	3	1	1	161.948	0.00E+00	-767313.1
'P1'	1	8	28	1	2	53.971	2500	34577.7



## Appendix C3

### HTHP Rich GC Reservoir: Pressure depletion

This appendix contains a portion of the complete data set for this example. It is from a North Sea full-field BO model containing a highly under-saturated, rich gas condensate. Two horizontal wells are used to produce this satellite structure.

The special feature of this case is the double BO→EOS6→EOS17 conversion. This first conversion uses BOz method and the second one used the generic conversion matrix, but again using pressure dependence for the split factors. This example shows the possibility of a consistent conversion from Black-oil simulation results to detailed Process component molar rates where profiles for individual components ( $C_1$  to  $C_6$ ) and heavy ends ( $C_{7+}$ ) are provided for every time step. This detailed information is valued for process simulation for the design of condensate recovery plants and was, in fact, useful for the economic evaluation of this satellite structure.

The complete driver file is included in this appendix. To conserve space and still include a complete ready-to-run data set, only a portion of the conversion file is listed. This portion corresponds to the portion of the input stream file included in this appendix.

The complete data set is included in the accompanying CD-Rom.

**Streamz Driver File: crit-gc-bo-6c.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:
INCLUDE 'bo.chr'

;Open the stream file containing streams with input characteriation:
DEFINE inp_str_file = 'crit-gc.str'
STREAMFILE INP1: INPUT ?inp_str_file?

;Include file containing the output characterization and the
;conversion definitions:
INCLUDE 'crit-gc.cnv'

;Open the output stream file to contain streams with output
;characteriation:
DEFINE out_str_file = 'Results\crit-gc-6c.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?

COPY
```

**Streamz Driver File: crit-gc-bo-6c-17c.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:
INCLUDE '6COMP.chr'

;Open the input stream file with input characterization:
DEFINE inp_str_file = 'Results\crit-gc-6c.str'
STREAMFILE INP1: INPUT ?inp_str_file?

;Include file containing the output characterization and the
;conversion definitions:
INCLUDE '6c-17c.cnv'

;Open the output stream file to contain streams with output
;characteriation:
DEFINE out_str_file = 'Results\crit-gc-6c-17c.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?

COPY
```

**Streamz Driver File: crit-gc-bo-6c-17c.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:
INCLUDE '6COMP.chr'

;Open the input stream file with input characteriation:
DEFINE inp_str_file = 'Results\crit-gc-6c.str'
STREAMFILE INP1: INPUT ?inp_str_file?

;Open the output stream file with output characteriation:
DEFINE out_str_file = 'Results\crit-gc-6c-tab.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?

;Include file containing other manipulations:
INCLUDE 'tab-6c.inc'
```

**Streamz Driver File: crit-gc-bo-6c-17c.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:
INCLUDE '17COMP.chr'

;Open the input stream file with input characteriation:
DEFINE inp_str_file = 'Results\crit-gc-6c-17c.str'
STREAMFILE INP1: INPUT ?inp_str_file?

;Open the output stream file streams with output characteriation:
DEFINE out_str_file = 'Results\crit-gc-6c-17c-tab.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?

;Include file containing other manipulations:
INCLUDE 'tab-17c.inc'
```

## Characterization Include File: bo.chr

CHAR 'BO\_Char'  
 NAME  
 SO  
 SG

**Conversion File: crit-gc.cnv**

TITLE 'Extendedblack-oil PVT generatedby PVTx'

CHAR 'crit-gc EOS6'

EOS PR

COMP	MW	TC K	PC BARA	AF	VS	AMOD	BMOD
X1	18.64	206.95	42.5	2.80E-2	0.00E+0	1	1
X2	58.89	426.65	35.2	0.184	0.00E+0	1	1
X3	112.42	618.15	28	3.32E-1	0.00E+0	1	1
CN1	179.98	698.15	21.3	0.44	0.00E+0	1	1
CN2	310	843.15	17.3	0.54	0.00E+0	1	1
CN3	480	1198.15	15	0.65	0.00E+0	1	1

BIPS	X1	X2	X3	CN1	CN2	CN3
X1	0.00E+0	0.00E+0	1.87E-2	6.50E-2	7.00E-2	7.00E-2
X2	0.00E+0	0.00E+0	2.10E-3	2.15E-2	2.15E-2	2.15E-2
X3	1.87E-2	2.10E-3	0.00E+0	0.00E+0	0.00E+0	0.00E+0
CN1	6.50E-2	2.15E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
CN2	7.00E-2	2.15E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
CN3	7.00E-2	2.15E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0

CONVERT 'BO\_Char' from VOLUMES to MOLES

SET	PVTNUM	1	PRES	(BARA)	422.073
SPLIT	SO	X1			
			3.66E-3	8.28E-1	1.01E+0
			2.15E-1		9.36E-1
					1.28E+0
SPLIT	SG	X1			
			3.84E-2	4.52E-3	1.06E-3
					2.37E-4
					-5.15E-4
					-2.35E-4
SET	PVTNUM	1	PRES	(BARA)	420
SPLIT	SO	X1			
			1.94E-3	8.32E-1	1.01E+0
			2.11E-1		9.39E-1
					1.28E+0
SPLIT	SG	X1			
			3.84E-2	4.51E-3	1.05E-3
					2.32E-4
					-5.17E-4
					-2.29E-4
SET	PVTNUM	1	PRES	(BARA)	415
SPLIT	SO	X1			
			-2.06E-3	8.43E-1	1.02E+0
			2.01E-1		9.46E-1
					1.28E+0
SPLIT	SG	X1			

```

      3.84E-2  4.50E-3  1.04E-3  2.21E-4  -5.20E-4
      -2.14E-4
SET      PVTNUM    1          PRES      (BARA)    410
SPLIT   SO        X1
      -6.16E-3  8.53E-1  1.03E+0  9.54E-1  1.28E+0
      1.92E-1
SPLIT   SG        X1
      3.84E-2  4.48E-3  1.02E-3  2.10E-4  -5.22E-4
      -2.00E-4

SET      PVTNUM    1          PRES      (BARA)    400
SPLIT   SO        X1
      -1.44E-2  8.74E-1  1.06E+0  9.68E-1  1.28E+0
      1.76E-1
SPLIT   SG        X1
      3.84E-2  4.45E-3  9.89E-4  1.89E-4  -5.24E-4
      -1.75E-4

```

### Characterization Include File: 17Comp.chr

```

CHAR      17COMP_CND
Comp      Mw          Tc          Pc
          K          BARA
N2        28.01      -146.95    33.94
CO2       44.01       31.05     73.76
C1        16.04      -82.55     46
C2        30.07       32.25     48.84
C3        44.1       96.65     42.46
IC4       58.12      134.95    36.48
NC4       58.12      152.05     38
IC5       71.837     193.98    35.247
NC5       7.22E+01   196.45    3.37E+01
C6        84.198     2.51E+02  34.7
C7        9.75E+01   285.07    3.21E+01
C8        1.11E+02   313.35    2.98E+01
C9        1.26E+02   337.1     2.75E+01
C10       1.39E+02   361.88    2.60E+01
CN1       180        415       21.9
CN2       310        550       16.3
CN3       480        925       15.9
end

```

### Conversion File: crit-gc.cnv

Include 17COMP.chr

Convert 'crit-gc EOS6' from Moles to Moles conserving Moles

Gamma X3 C7 file GAM1

AVE = 1

Set Pres 423 (bara)

```

SPLIT   X1      CO2  0.03514
SPLIT   X1      N2   0.00462
SPLIT   X1      C1   0.84342
SPLIT   X1      C2   0.11682
SPLIT   X2      C3   0.50204

```

SPLIT	X2	IC4	0.07055
SPLIT	X2	NC4	0.19427
SPLIT	X2	IC5	0.05624
SPLIT	X2	NC5	0.08078
SPLIT	X2	C6	0.09611

**Extra Manipulation Include Files: tab-6c.inc**

```
domain time t1 t2
lump tot 6*1
filter prod tot moles gt 0
tabulate time days if prod
```

**Extra Manipulation Include Files: tab-17c.inc**

```
domain time t1 t2
lump tot 17*1
filter prod tot moles gt 0
tabulate time days if prod
```

**Extract of Main Stream File: ff-in.str**

```

STREAMZ          1
NOTE             'Stream file created by PSM-Toolkit'
NOTE             'Black-Oil Characterization'
NOTE             'Unit system: METRIC'
NOTE             'Calculation level: CONNECTION'
CHAR             'BO_Char'
VARIABLE FIELD   STRING
VARIABLE GROUP   STRING
VARIABLE WELL    STRING
VARIABLE PVTNUM  INTEGER
VARIABLE CONN_I  INTEGER
VARIABLE CONN_J  INTEGER
VARIABLE CONN_K  INTEGER
VARIABLE TSTEP   INTEGER
VARIABLE T1      TIME
VARIABLE T2      TIME
VARIABLE PFLAG   INTEGER
VARIABLE PRES    PRESSURE
DATA

SET              TSTEP          3
SET              T1              71.5 (days)
SET              T2              110 (days)

WELL  CONN_I  CONN_J  CONN_K  PVTNUM  PRES (bara)  VOLUME SO (Sm3/day)  VOLUME SG (Sm3/day)
'H04_S21'  17    29     9       9       581.868      347.7487      222297.4
'H04_S21'  18    30    10      10      585.161      525.8461      336145.6
'H04_S21'  19    31    13      13      576.726      269.1743      172068.9
'H04_S21'  20    31    13      13      575.22       180.6915      115506.5
'H04_S21'  21    32    13      13      572.77       220.1713      140743.9
'H04_S21'  22    32    13      13      422.073      180.7744      115559.5

```

'H04_S21'	23	33	9	9	569.528	132.584	84753.97
'H04_S21'	24	33	8	8	569.287	110.8929	70887.99
'H03_S23'	18	36	10	10	422.073	505.7763	323316.1
'H03_S23'	19	36	13	13	578.877	228.3766	145989.1
'H03_S23'	20	36	18	18	422.073	539.2634	344722.6
'H03_S23'	22	38	13	13	422.073	219.0202	140008.1
'H03_S23'	23	39	10	10	579.458	333.2425	213024.4
'H03_S23'	24	40	8	8	422.073	203.6464	130180.4

SET TSTEP 4  
 SET T1 110 (days)  
 SET T2 149.419 (days)

WELL	CONN_I	CONN_J	CONN_K	PVTNUM	PRES (bara)	VOLUME SO (Sm <sup>3</sup> /day)	VOLUME SG (Sm <sup>3</sup> /day)
'H04_S21'	17	29	9	9	557.869	379.4358	242553.3
'H04_S21'	18	30	10	10	422.073	571.0797	365061.1
'H04_S21'	19	31	13	13	552.053	281.5626	179988.1
'H04_S21'	20	31	13	13	550.43	184.6112	118012.2
'H04_S21'	21	32	13	13	422.073	206.8512	132229.1
'H04_S21'	22	32	13	13	422.073	168.0466	107423.3
'H04_S21'	23	33	9	9	543.923	99.93812	63885.16
'H04_S21'	24	33	8	8	543.693	82.10536	52485.63
'H03_S23'	18	36	10	10	562.836	541.087	345888.4
'H03_S23'	19	36	13	13	554.502	283.5606	181265.3
'H03_S23'	20	36	18	18	551.077	543.6639	347535.6
'H03_S23'	22	38	13	13	550.765	218.1359	139442.8
'H03_S23'	23	39	10	10	422.073	272.2851	174057.5
'H03_S23'	24	40	8	8	551.151	160.9994	102918.4



## Appendix C4

### Rich GC with Compositional Gradient

This appendix contains a portion of the complete data set for this example. This is from a North Sea 3D Sector model containing near-critical fluid with compositional gradient. 3 producing wells located in upper-, middle-, and lower structure.

The special feature of this case is compositional gradient resulting in a different initial fluid composition varying with depth. Black oil simulation results are post-processed by the BOz method. Conversions are performed at grid-cell level thereby capturing the correct GOR and converting it to the correct compositions. Compositions (molar rates) are then summed to well-level for reporting.

The complete driver file is included in this appendix. To conserve space and still include a complete ready-to-run data set, only a portion of the conversion file is listed. This portion corresponds to the portion of the input stream file included in this appendix.

The complete data set is included in the accompanying CD-Rom.

**Streamz Driver File: bo-eos6-eos22\_3w.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:
INCLUDE 'bo6_3w.chr'

;Open the input stream file with input characteriation:
DEFINE inp_str_file = 'bo6_3w.str'
STREAMFILE INP1: INPUT ?inp_str_file?

;Include file containing the output characterization and the
;conversion definitions:
INCLUDE 'eos6_3w.cnv'

;Open the output stream file streams with output characteriation:
DEFINE out_str_file = 'eos6_3w_a.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?

;Include file containing other manipulations:
INCLUDE 'bo-eos6-eos22_3w.inc'
```

**Characterization Include File: bo6\_3w.chr**

CHAR 'BO\_Char'  
 NAME  
 SO  
 SG

**Conversion File: eos6\_3w.cnv**

TITLE 'Extendedblack-oilPVT generatedby PVTx'

CHAR '6COMP'

EOS SRK

COMP	MW	TC K	PC BARA	AF	VS	AMOD	BMOD
C1N2	16.0784	190.267	45.9393	8.17E-3	2.29E-2	0.998749	0.99743
CO2C2	35.4492	304.825	60.7803	0.158844	6.67E-2	1.03857	1.05558
C3-6	55.1443	418.897	37.8228	1.97E-1	9.92E-2	0.984345	0.96603
C7-9F1-2	116.912	577.393	28.3558	0.543183	1.09E-1	0.988337	1.03223
F3-8	280.992	753.268	15.1953	0.973685	1.18E-1	0.968595	0.95487
F9	621.628	979.286	12.0641	1.32405	-1.34E-1	1	1

BIPS	C1N2	CO2C2	C3-6	C7-9F1-2	F3-8	F9
C1N2	0.00E+0	5.74E-2	4.13E-4	2.72E-4	2.72E-4	2.72E-4
CO2C2	5.74E-2	0.00E+0	5.75E-2	4.79E-2	4.79E-2	4.79E-2
C3-6	4.13E-4	5.75E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
C7-9F1-2	2.72E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F3-8	2.72E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F9	2.72E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0

CONVERT 'BO\_Char' from VOLUMES to MOLES

SET PVTNUM 1 PRES (BARA) 465.329

SPLIT SO C1N2  
 -4.17E-1 2.87E-1 6.03E-1 1.04E+0 1.44E+0  
 4.92E-1

SPLIT SG C1N2  
 3.18E-2 7.19E-3 3.36E-3 1.93E-3 -9.49E-5  
 -3.93E-4

SET PVTNUM 1 PRES (BARA) 450

SPLIT SO C1N2  
 -4.21E-1 2.87E-1 6.05E-1 1.05E+0 1.45E+0  
 4.88E-1

SPLIT SG C1N2  
 3.18E-2 7.19E-3 3.35E-3 1.92E-3 -1.06E-4  
 -3.85E-4

SET PVTNUM 1 PRES (BARA) 425

SPLIT SO C1N2  
 -4.29E-1 2.87E-1 6.12E-1 1.07E+0 1.47E+0  
 4.75E-1

SPLIT SG C1N2  
 3.18E-2 7.19E-3 3.34E-3 1.88E-3 -1.47E-4  
 -3.56E-4

SET PVTNUM 1 PRES (BARA) 400

```

SPLIT SO C1N2
      -4.36E-1 2.87E-1 6.20E-1 1.09E+0 1.48E+0
      4.61E-1
SPLIT SG C1N2
      3.18E-2 7.18E-3 3.32E-3 1.83E-3 -1.99E-4
      -3.16E-4

SET PVTNUM 1 PRES (BARA) 375
SPLIT SO C1N2
      -4.42E-1 2.86E-1 6.27E-1 1.11E+0 1.50E+0
      4.49E-1
SPLIT SG C1N2
      3.18E-2 7.18E-3 3.29E-3 1.77E-3 -2.49E-4
      -2.74E-4

```

### Characterization Include File: eos22.chr

EOS SRK

```

CHAR 22COMP
COMP
N2
CO2
C1
C2
C3
IC4
NC4
IC5
NC5
C6
C7
C8
C9
F1
F2
F3
F4
F5
F6
F7
F8
F9

```

### Conversion File: eos6-eos22.cnv

Include 'eos22.chr'

```

Convert 6COMP from MOLES
Set C7PLUS MOLES/MOLE 0.134427371
Split C1N2 N2 0.002767442
Split CO2C2 CO2 0.380573441
Split C1N2 C1 0.997232558
Split CO2C2 C2 0.619426559
Split C3-6 C3 0.506522522
Split C3-6 IC4 0.089045004
Split C3-6 NC4 0.170697388
Split C3-6 IC5 0.062416609
Split C3-6 NC5 0.069379161
Split C3-6 C6 0.101939315

```

Split	C7-9F1-2	C7	0.232139504
Split	C7-9F1-2	C8	0.275194685
Split	C7-9F1-2	C9	0.174448467
Split	C7-9F1-2	F1	0.174174655
Split	C7-9F1-2	F2	0.14404269
Split	F3-8	F3	0.18315682
Split	F3-8	F4	0.220960674
Split	F3-8	F5	0.117507342
Split	F3-8	F6	0.18307029
Split	F3-8	F7	0.131345026
Split	F3-8	F8	0.163959848
Split	F9	F9	1
Set	C7PLUS	MOLES/MOLE	0.124249052
Split	C1N2	N2	0.003018637
Split	CO2C2	CO2	0.369382019
Split	C1N2	C1	0.996981363
Split	CO2C2	C2	0.630617981
Split	C3-6	C3	0.512456322
Split	C3-6	IC4	0.089267578
Split	C3-6	NC4	0.169929926
Split	C3-6	IC5	0.061442449
Split	C3-6	NC5	0.068002116
Split	C3-6	C6	0.098901609
Split	C7-9F1-2	C7	0.233060794
Split	C7-9F1-2	C8	0.275511201
Split	C7-9F1-2	C9	0.174276174
Split	C7-9F1-2	F1	0.173718438
Split	C7-9F1-2	F2	0.143433393
Split	F3-8	F3	0.186081857
Split	F3-8	F4	0.223705068
Split	F3-8	F5	0.118470338
Split	F3-8	F6	0.183264746
Split	F3-8	F7	0.12978304
Split	F3-8	F8	0.158694951
Split	F9	F9	1

**Extra Manipulation Include File: bo-eos6-eos22\_3w.inc**

```

TITLE 'Conversion to 6-Component EOS'

DOMAIN TIME T1 T2
FILTER VALID: PRES GT 0 PSIA

TABULATE TSTEP and TIME (days) and WELL if VALID

STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE

STREAMFILE INP1 INPUT 'eos6_3w_a.str'
STREAMFILE OUT1 OUTPT 'Results\eos6_3w_b.str'

FILTER WELL: WELL EQ 'P5010', COPY IF WELL
FILTER WELL: WELL EQ 'P2505', COPY IF WELL
FILTER WELL: WELL EQ 'P0101', COPY IF WELL

TABULATE TSTEP AND TIME (days)

STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE

STREAMFILE INP1 INPUT 'Results\eos6_3w_b.str'

LUMP C7PLUS = C7-9F1-2, F3-8, F9

INCLUDE eos22.chr

;The above file is included because the following file is
;generated by STR2CNV and doesnot include a characterization

INCLUDE 'eos6-eos22.cnv'

STREAMFILE OUT1 OUTPT 'eos22_3w_a.str'

COPY

STREAMFILE INP1 CLOSE
STREAMFILE OUT1 CLOSE

STREAMFILE OUT1 OUTPT 'Results\eos22_3w_b.str', VARIABLE LABEL
STRING

LUMP TOTAL 22*1

DEFINE INFILE 'eos22_3w_a.str'

DEFINE NAME 'P5010', INCLUDE 'wellsums.inc'
DEFINE NAME 'P2505', INCLUDE 'wellsums.inc'
DEFINE NAME 'P0101', INCLUDE 'wellsums.inc'
DEFINE NAME '', INCLUDE 'wellsums.inc'

```

**Extra Manipulation Include File: wellsums.inc**

```

STREAMFILE INP1 INPUT '?INFILE?'

DEFINE TEMPFILE 'streamz_temp.str'
STREAMFILE TMP1 OUTPUT '?TEMPFILE?'

FILTER WELL: WELL EQ '?NAME?', COPY TO TMP1 IF WELL

```

```

STREAMFILE INP1 CLOSE
STREAMFILE TMP1 CLOSE

STREAMFILE INP1 INPUT '?TEMPFILE?'

DEFINE T2 '0.0'
DEFINE T1 '?T2?', DEFINE T2 '0.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '1.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '1.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '2.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '2.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '3.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '3.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '4.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '4.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '5.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '5.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '6.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '6.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '7.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '7.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '8.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '8.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '9.0', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '9.5', INCLUDE timesums.inc
DEFINE T1 '?T2?', DEFINE T2 '10.0', INCLUDE timesums.inc

WRITE STREAMS YEARS_0.0-0.5
          AND YEARS_0.5-1.0
          AND YEARS_1.0-1.5
          AND YEARS_1.5-2.0
          AND YEARS_2.0-2.5
          AND YEARS_2.5-3.0
          AND YEARS_3.0-3.5
          AND YEARS_3.5-4.0
          AND YEARS_4.0-4.5
          AND YEARS_4.5-5.0
          AND YEARS_5.0-5.5
          AND YEARS_5.5-6.0
          AND YEARS_6.0-6.5
          AND YEARS_6.5-7.0
          AND YEARS_7.0-7.5
          AND YEARS_7.5-8.0
          AND YEARS_8.0-8.5
          AND YEARS_8.5-9.0
          AND YEARS_9.0-9.5
          AND YEARS_9.5-10.0

STREAMFILE INP1 CLOSE
CLEAR

```

### Extra Manipulation Include File: timesums.inc

```

FILTER ACTIVE: TOTAL MOLES NE 0.0
FILTER YEAR: TIME GE ?T1? YEARS AND TIME LE ?T2? YEARS AND ACTIVE

COMBINE YEARS_?T1?-?T2? IF YEAR, WEIGHTING OVER TIME (days)

TAG YEARS_?T1?-?T2?: LABEL = 'Years ?T1? to ?T2?'

```

**Extract of Main Stream File: bo6\_3w.str.str**

```

STREAMZ 1
NOTE 'Stream file created by PSM-Toolkit'
NOTE 'Black-Oil Characterization'
NOTE 'Unit system: METRIC'
NOTE 'Calculation level: CONNECTION'
CHAR 'BO_Char'
VARIABLE FIELD STRING
VARIABLE GROUP STRING
VARIABLE WELL STRING
VARIABLE PVTNUM INTEGER
VARIABLE CONN_I INTEGER
VARIABLE CONN_J INTEGER
VARIABLE CONN_K INTEGER
VARIABLE TSTEP INTEGER
VARIABLE T1 TIME
VARIABLE T2 TIME
VARIABLE PFLAG INTEGER
VARIABLE PRES PRESSURE
DATA

SET TSTEP 1
SET T1 0 (days)
SET T2 0.01 (days)

WELL CONN_I CONN_J CONN_K PVTNUM PFLAG PRES (bara) VOLUME SO (Sm3/day) VOLUME SG (Sm3/day)
'P0101' 1 1 10 1 1 377.999 24.10581 36907.13
'P0101' 1 1 9 1 1 377.83 24.11837 36966.1
'P0101' 1 1 8 1 1 377.593 24.07373 36953.59
'P0101' 1 1 7 1 1 377.263 24.11332 37092.71
'P0101' 1 1 6 1 1 376.81 24.17399 37294.19
'P0101' 1 1 5 1 1 376.195 24.22832 37526.38

```

---

'P0101' 1	1	4	1	1	375.346	24.55738	38245.42
'P0101' 1	1	3	1	1	373.974	25.16124	39497.34
'P0101' 1	1	2	1	1	371.732	26.46545	42067.82
'P0101' 1	1	1	1	1	366.554	30.89463	50579.05
'P2505' 25	5	10	1	1	418.103	30.33817	36518.82
'P2505' 25	5	9	1	1	417.766	30.34606	36598.68
'P2505' 25	5	8	1	1	417.283	30.27135	36609.99
'P2505' 25	5	7	1	1	416.608	30.28148	36764.81
'P2505' 25	5	6	1	1	415.687	30.27925	36958.33
'P2505' 25	5	5	1	1	414.475	30.20066	37123.37
'P2505' 25	5	4	1	1	412.864	30.34334	37653.03
'P2505' 25	5	3	1	1	410.676	30.61435	38485.52
'P2505' 25	5	2	1	1	407.378	31.3537	40206.87
'P2505' 25	5	1	1	1	400.113	34.1065	45762.45
'P5010' 50	10	10	1	2	427.991	76.66275	24118.55
'P5010' 50	10	9	1	2	428.22	76.88937	24227.56
'P5010' 50	10	8	1	2	428.549	77.01872	24322.77
'P5010' 50	10	7	1	2	429.044	77.55083	24573.08
'P5010' 50	10	6	1	2	429.83	78.36221	24962.19
'P5010' 50	10	5	1	2	430.959	79.4131	25489.31
'P5010' 50	10	4	1	2	432.577	81.72298	26514.21
'P5010' 50	10	3	1	2	435.045	85.47847	28185.12
'P5010' 50	10	2	1	2	439.34	92.45949	31338.52
'P5010' 50	10	1	1	2	465.636	80.73639	34273.43



## Appendix C5

### Rich GC Reservoir with Constant Composition

This appendix contains a portion of the complete data set for this example. This is from a North Sea 3D Sector model containing near-critical fluid with constant composition. Gas injection takes place from up-dip well and production from down-dip well. Black oil simulation results are post-processed by the BOz method. Conversions are performed at grid-cell level thereby capturing the correct GOR and converting it to the correct compositions. Compositions (molar rates) are then summed to well-level. Conversions from 6-component molar rates are performed at well level with split factors a function of the well stream  $C_{6+}$  mole fraction.

The complete driver file is included in this appendix. To conserve space and still include a complete ready-to-run data set, only a portion of the conversion file is listed. This portion corresponds to the portion of the input stream file included in this appendix.

The complete data set is included in the accompanying CD-Rom.

**Streamz Driver File: bo6-eos6.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:
INCLUDE 'bo6.chr'

;Open the input stream file with input characteriation:
DEFINE inp_str_file = 'Results\bo6.str'
STREAMFILE INP1: INPUT ?inp_str_file?

;Include file containing the output characterization and the
;conversion definitions:

INCLUDE 'eos6.cnv'

;Open the output stream file with output characteriation:
DEFINE out_str_file = 'bo6-eos6.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?
;Include file containing other manipulations:
INCLUDE 'tabulate.inc'
```

**Streamz Driver File: eos6-eos22.stz**

```
TITLE 'Streamz input file generated by PreStreamz'

;Include file containing the input characterization:
INCLUDE 'eos6.chr'

LUMP C7PLUS C7-9F1-2 F3-8 F9

;Open the input stream file with input characteriation:
DEFINE inp_str_file = 'bo6-eos6.str'
STREAMFILE INP1: INPUT ?inp_str_file?

;Include file containing the output characterization and the
;conversion definitions:
INCLUDE 'eos6-eos22.cnv'

;Open the output stream file streams with output characteriation:
DEFINE out_str_file = 'Results\bo6-eos6-eos22.str'
STREAMFILE OUT1: OUTPUT ?out_str_file?

;Include file containing other manipulations:
INCLUDE 'tabulate.inc'
```

**Characterization Include File: bo6.chr**

```
CHAR 'BO_Char'
NAME
SO
SG
```

**Conversion File: eos6\_3w.cnv**

```
CHAR '6COMP'
```

```
EOS SRK
```

COMP	MW	TC K	PC BARA	AF	VS	AMOD	BMOD
C1N2	16.0784	190.267	45.9393	8.17E-3	2.29E-2	0.998749	0.997437
CO2C2	35.4492	304.825	60.7803	1.59E-1	6.67E-2	1.03857	1.05558
C3-6	55.1443	418.897	37.8228	0.196838	9.92E-2	0.984345	0.966035
C7-9F1-2	116.912	577.393	28.3558	5.43E-1	1.09E-1	0.988337	1.03223
F3-8	280.992	753.268	15.1953	0.973685	1.18E-1	0.968595	0.954872
F9	621.628	979.286	12.0641	1.32405	-1.34E-1	1	1

BIPS	C1N2	CO2C2	C3-6	C7-9F1-2	F3-8	F9
C1N2	0.00E+0	5.74E-2	4.13E-4	2.72E-4	2.72E-4	2.72E-4
CO2C2	5.74E-2	0.00E+0	5.75E-2	4.79E-2	4.79E-2	4.79E-2
C3-6	4.13E-4	5.75E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
C7-9F1-2	2.72E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F3-8	2.72E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F9	2.72E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0

```
CONVERT 'BO_Char' from VOLUMES to MOLES
```

SET	PVTNUM	1	PFLAG	2	PRES	(BARA)	465.81
SPLIT	SO	C1N2					
			-4.31E-1	2.92E-1	6.30E-1	1.04E+0	1.51E+0
			4.59E-1				
SPLIT	SG	C1N2					
			3.18E-2	7.10E-3	3.41E-3	1.88E-3	-1.14E-4
			-3.72E-4				

SET	PVTNUM	1	PFLAG	1	PRES	(BARA)	465.81
SPLIT	SO	C1N2					
			-3.81E+0	2.92E+0	1.55E+0	2.61E+0	1.41E+0
			1.35E-1				
SPLIT	SG	C1N2					
			3.57E-2	4.07E-3	2.35E-3	7.60E-5	0.00E+0
			0.00E+0				

**Characterization Include File: eos6.chr**

```
EOS 'SRK'
```

CHAR	'6comp'						
COMP	MW	TC K	PC BARA	AF	VS	AMOD	BMOD
C1N2	16.0784	190.267	45.9393	8.17E-3	2.29E-2	0.99874	0.997457
CO2C2	35.4492	304.825	60.7803	0.15884	6.67E-2	1.03857	1.05563
C3-6	55.1443	418.897	37.8228	1.97E-1	9.92E-2	0.98434	0.966063
C7-9F1-2	116.912	577.393	28.3558	0.54318	1.09E-1	0.98833	1.0322
F3-8	280.992	753.268	15.1953	0.97369	1.18E-1	0.96859	0.954867
F9	621.628	979.286	12.0641	1.32405	-1.34E-1	1	0.999996

END

BIPS	C1N2	CO2C2	C3-6	C7-9F1-2	F3-8	F9
C1N2	0.00E+0	5.74E-2	4.10E-4	2.70E-4	2.70E-4	2.70E-4
CO2C2	5.74E-2	0.00E+0	5.75E-2	4.79E-2	4.79E-2	4.79E-2
C3-6	4.10E-4	5.75E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
C7-9F1-2	2.70E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F3-8	2.70E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F9	2.70E-4	4.79E-2	0.00E+0	0.00E+0	0.00E+0	0.00E+0

**Characterization Include File: eos22.chr**

EOS SRK

CHAR 22COMP

COMP

N2

CO2

C1

C2

C3

IC4

NC4

IC5

NC5

C6

C7

C8

C9

F1

F2

F3

F4

F5

F6

F7

F8

F9

**Conversion File: eos6-eos22.cnv**

Include 'eos22.chr'

Convert	6COMP	from	MOLES	to	MOLES
Set	C7PLUS	MOLES/MOLE	0.101558		
Split	C1N2	N2	0.002896		
Split	CO2C2	CO2	0.38407		
Split	C1N2	C1	0.997104		
Split	CO2C2	C2	0.61593		
Split	C3-6	C3	0.517121		
Split	C3-6	IC4	0.08938		
Split	C3-6	NC4	0.169192		
Split	C3-6	IC5	0.060743		
Split	C3-6	NC5	0.067072		
Split	C3-6	C6	0.096493		
Split	C7-9F1-2	C7	0.240523		
Split	C7-9F1-2	C8	0.279631		
Split	C7-9F1-2	C9	0.173842		
Split	C7-9F1-2	F1	0.169568		
Split	C7-9F1-2	F2	0.136436		
Split	F3-8	F3	0.209153		

---

Split	F3-8	F4	0.24115
Split	F3-8	F5	0.122641
Split	F3-8	F6	0.179793
Split	F3-8	F7	0.118108
Split	F3-8	F8	0.129155
Split	F9	F9	1
Set	C7PLUS	MOLES/MOLE	0.09147
Split	C1N2	N2	0.003253
Split	CO2C2	CO2	0.364529
Split	C1N2	C1	0.996747
Split	CO2C2	C2	0.635471
Split	C3-6	C3	0.526279
Split	C3-6	IC4	0.089754
Split	C3-6	NC4	0.168095
Split	C3-6	IC5	0.059235
Split	C3-6	NC5	0.064918
Split	C3-6	C6	0.09172
Split	C7-9F1-2	C7	0.241689
Split	C7-9F1-2	C8	0.279826
Split	C7-9F1-2	C9	0.173487
Split	C7-9F1-2	F1	0.16903
Split	C7-9F1-2	F2	0.135967
Split	F3-8	F3	0.209826
Split	F3-8	F4	0.241774
Split	F3-8	F5	0.122861
Split	F3-8	F6	0.179864
Split	F3-8	F7	0.117812
Split	F3-8	F8	0.127863
Split	F9	F9	1

**Extra Manipulation Include File: tabulate.inc**

```
domain time t1 t2
filter prod well sw 'P'
tabulate well and time days if prod
```

**Extract of Main Stream File: bo6.str**

```

STREAMZ  1
NOTE     'Stream file created by PSM-Toolkit'
NOTE     'Black-Oil Characterization'
NOTE     'Unit system: METRIC'
NOTE     'Calculation level: CONNECTION'
CHAR     'BO_Char'
VARIABLE FIELD  STRING
VARIABLE GROUP  STRING
VARIABLE WELL   STRING
VARIABLE PVTNUM INTEGER
VARIABLE CONN_I INTEGER
VARIABLE CONN_J INTEGER
VARIABLE CONN_K INTEGER
VARIABLE TSTEP  INTEGER
VARIABLE T1     TIME
VARIABLE T2     TIME
VARIABLE PFLAG  INTEGER
VARIABLE PRES   PRESSURE
DATA

```

```

SET      TSTEP  1
SET      T1     0      (days)
SET      T2     30.674 (days)

```

WELL	CONN_I	CONN_J	CONN_K	PVTNUM	PFLAG	PRES (bara)	VOLUME SO (Sm <sup>3</sup> /day)	VOLUME SG (Sm <sup>3</sup> /day)
'I0101'	1	5	10	1	1	497.007	0.00E+00	-143335
'I0101'	1	5	9	1	1	496.991	0.00E+00	-143565
'I0101'	1	5	8	1	1	496.969	0.00E+00	-143444
'I0101'	1	5	7	1	1	496.939	0.00E+00	-143738
'I0101'	1	5	6	1	1	496.896	0.00E+00	-143917

---

'I0101'	1	5	5	1	1	496.838	0.00E+00	-143540
'I0101'	1	5	4	1	1	496.757	0.00E+00	-143707
'I0101'	1	5	3	1	1	496.638	0.00E+00	-142447
'I0101'	1	5	2	1	1	496.44	0.00E+00	-132931
'I0101'	1	5	1	1	1	495.82	0.00E+00	-98817.2
'P1505'	50	5	10	1	1	490.874	128.6183	111834.3
'P1505'	50	5	9	1	1	490.85	129.298	112425.2
'P1505'	50	5	8	1	1	490.816	129.1957	112336.3
'P1505'	50	5	7	1	1	490.768	129.4612	112567.1
'P1505'	50	5	6	1	1	490.702	129.6037	112691.1
'P1505'	50	5	5	1	1	490.613	129.2124	112350.8
'P1505'	50	5	4	1	1	490.489	129.336	112458.3
'P1505'	50	5	3	1	1	490.312	129.1653	112309.8
'P1505'	50	5	2	1	1	490.035	129.3775	112494.3
'P1505'	50	5	1	1	1	489.325	129.3311	112454

## Appendix C6

### TBP stream data Fit

This appendix contains the complete data set for this example. The main purpose here is the determination of Gamma distribution parameters, which can later be used to split C7+ fractions into user specified number of fractions.

This data set is also included in the accompanying CD-Rom.

**Streamz Driver File: TBP-SCN.stz**

```
ECHO ON
title 'FIT TBP and split to K-F SCN'
```

```
char 'TBP'
comp      MW
C7S      96
C8S      101
C9S      114
C10S     129
C11S     144
C12S     163
C13S     177
C14S     192
C15+     330
END
```

```
splitfile spl1 open GetSplits.spl
gammafile gam1 open GetSplits.gam
```

```
CHAR 'SCN'
COMPS    MW    LMW
C7       96
C8       107
C9       121
C10      134
C11      147
C12      161
C13      175
C14      190
C15      206
C16      222
C17      237
C18      256
C19      263
C20      275
C21      291
C22      305
C23      318
C24      331
C25      345
C26      359
C27      374
C28      388
C29      402
C30      416
C31      430
C32      444
C33      458
C34      472
C35      486
C36      500
C37      514
C38      528
C39      542
C40      556
C41      570
C42      584
C43      598
C44      612
C45      619
END
```

---

```
streamfile out output tbp-scn.str  
  
convert TBP from MOLES to MOLES conserving MASS  
splits spl1  
gamma C7S C7 file gam1  
  
copy
```

**Main Stream File: tbp.str**

STREAMZ 1

Char 'TBP'

DATA

MOLES

1.12	1.3	1.18	0.98	0.62	0.57	0.74	0.53	4.1
------	-----	------	------	------	------	------	------	-----

## Appendix C7

### Splitting of a Single C<sub>7+</sub> fraction.

This appendix contains the complete data set for this example. Gamma distribution parameters have previously been obtained from a TBP analysis of a related sample and the number of plus fractions and their average MW have been fixed. Streamz is used to convert the single fraction stream to the desired characterization. The pre-determined model is used to split into the desired discrete characterization with user defined average molecular weights.

This data set is also included in the accompanying CD-Rom.

**Streamz Driver File: C7+Split.stz**

```
ECHO ON
title 'Split C7+'

char 'C7+'
comp      MW
C7+      203.779
END

streamfile in input C7+.str

splitfile spl1 open C7+.spl
gammafile gam1 open C7+.gam

char 'TBP'
comp      MW      ~LMW  LMW
C7S      92.91   90.27
C8S      102.25  96.58
C9S      115.68  108.65
C10S     130.72  123.33
C11S     144.25  138.61
C12S     156.33  150.15
C13S     172.00  162.79
C14S     189.49  181.78
C15+     334.48  197.57      197.57
END

streamfile out output tbp.str

convert C7+ from MOLES to MOLES ;conserving MASS
splits spl1
gamma C7+ C7s file gam1
Shape    0.72872
Average  203.779
Bound    90.267
Origin   1.00000

copy
```

**Main Stream File: tbp.str**

```
STREAMZ 1  
Char 'C7+'  
DATA
```

```
MOLES  
11.14
```



## Appendix C8

### Comparison of Exponential vs. Gamma Fit

This appendix contains the complete data set for the the Gamma Distribution portion of this example. Uwaint field came up with 10 different samples, to be used for development of an EOS model. The  $C_{7+}$  fraction for all samples had GC analysis. For 9 of the 10 samples,  $C_{7+}$  were split into 14 fractions while the last into 6.

For purpose of splitting the  $C_{7+}$  fraction into any required number of fractions it is usually fit to a continuous model. Fitting to a simple exponential model is possible using the procedure outline the SPE Monograph on Phase Behavior. This can be implemented in a spreadsheet. The same data is also fit using the Gamma distribution model. The superiority of the Gamma distribution over the exponential model is shown.

This data set is also included in the accompanying CD-Rom.

**Streamz Driver File: Uwainat.stz**

TITLE 'Uwainat EB-3H Well ECM Oil C7+ Distribution'

CHAR 'SAMPLE1+2'

Comp	MW
F7	96
F8	107
F9	121
F10	134
F11	147
F12	161
F13	175
F14	190
F15	206
F16	222
F17	237
F18	251
F19	263
F20+	408

Mix "Sample1"

F7 Moles	3.29	5.68	6.02	5.04	4.19	3.94	3.99	3.82
	3.91	3.78	3.45	3.46	3.73	41.64		

Mix "Sample2"

F7 Moles	1.46	1.97	1.84	1.47	1.21	1.13	1.14	1.10
	1.12	1.08	0.99	0.99	1.07	11.93		

end

GAMMAFILE GAM1 OPEN 'Uwainat.gam'

STREAMFILE OUT1 OUTPUT 'streamz output/Sample1+2.str'

end

CHAR SAMPLE3

Comp	MW
F7	96
F8	107
F9	121
F10	134
F11	147
F12	161
F13	175
F14	190
F15	206
F16	222
F17	237
F18	251
F19	263
F20+	311

Mix "Sample3"

F7 Moles	0.45	0.47	0.35	0.21	0.15	0.13	0.12	0.10
	0.08	0.07	0.05	0.05	0.04	0.14		

end

STREAMFILE OUT3 OUTPUT 'streamz output/Sample3.str'

end

CHAR SAMPLE4

Comp	MW
F7	96
F8	107

```

F9      121
F10     134
F11     147
F12     161
F13     175
F14     190
F15     206
F16     222
F17     237
F18     251
F19     263
F20+    426
End

```

```
Mix "Sample4"
```

```

F7 Moles 6.40  8.47  7.56  5.76  4.47  4.01  3.83  3.69
          3.32  3.23  2.90  2.81  3.00  28.89

```

```
end
```

```
STREAMFILE OUT4  OUTPUT 'streamz output/Sample4.str'
```

```
end
```

```
CHAR SAMPLE5
```

```

Comp      MW
F7        96
F8        107
F9        121
F10       134
F11       147
F12       161
F13       175
F14       190
F15       206
F16       222
F17       237
F18       251
F19       263
F20+     406
End

```

```
Mix "Sample5"
```

```

F7 Moles 6.06  8.18  7.33  5.67  4.41  3.90  3.73  3.42
          3.42  3.16  2.84  2.80  2.96  31.41

```

```
end
```

```
STREAMFILE OUT5  OUTPUT 'streamz output/Sample5.str'
```

```
end
```

```
CHAR SAMPLE6
```

```

Comp      MW
F7        96
F8        107
F9        121
F10       134
F11       147
F12       161
F13       175
F14       190
F15       206
F16       222
F17       237
F18       251
F19       263
F20+     402

```

End

Mix "Sample6"

F7 Moles	6.12	8.30	7.36	5.77	4.44	3.94	3.78	3.48
	3.42	3.21	2.86	2.81	2.98	31.19		

end

STREAMFILE OUT6 OUTPUT 'streamz output/Sample6.str'

end

CHAR SAMPLE7

Comp	MW
F7	96
F8	107
F9	121
F10	134
F11	147
F12	161
F13	175
F14	190
F15	206
F16	222
F17	237
F18	251
F19	263
F20+	395

End

Mix "Sample7"

F7 Moles	6.02	8.12	7.23	5.59	4.34	3.81	3.67	3.32
	3.42	3.11	2.79	2.74	2.93	32.21		

end

STREAMFILE OUT7 OUTPUT 'streamz output/Sample7.str'

end

CHAR SAMPLE8

Comp	MW
F7	96
F8	107
F9	121
F10	134
F11	147
F12	161
F13	175
F14	190
F15	206
F16	222
F17	237
F18	251
F19	263
F20+	416

End

Mix "Sample8"

F7 Moles	6.30	8.46	7.45	5.71	4.44	3.94	3.78	3.50
	3.40	3.20	2.87	2.81	3.08	30.00		

end

STREAMFILE OUT8 OUTPUT 'streamz output/Sample8.str'

end

CHAR SAMPLE9

Comp	MW
F7	96

```

F8      107
F9      121
F10     134
F11     147
F12     161
F13     175
F14     190
F15     206
F16     222
F17     237
F18     251
F19     263
F20+    415
End

Mix "Sample9"
F7 Moles 6.25  8.38  7.44  5.71  4.42  3.78  3.74  3.44
          3.40  3.16  2.84  2.80  2.96  30.12
end
STREAMFILE OUT9  OUTPUT 'streamz output/Sample9.str'
end

CHAR SAMPLE10
Comp    MW
F7      96
F8      107
F9      121
F10     134
F11     147
F20+    325
End

Mix "Sample10"
F7 Moles 4.35  6.55  6.01  4.86  3.70  69.17
end
STREAMFILE OUT10 OUTPUT 'streamz output/Sample10.str'
end

char Dum
Comp MW
Dum1 268
end

CONVERT Sample1+2 from MOLES to MOLES conserving MOLES
      GAMMA F7 Dum1 FILE GAM1
      ZERO 1.0 ;
end
CONVERT Sample3 from MOLES to MOLES conserving MOLES
      GAMMA F7 Dum1 FILE GAM1
      ZERO 1.0 ;
end
CONVERT Sample4 from MOLES to MOLES conserving MOLES
      GAMMA F7 Dum1 FILE GAM1
      ZERO 1.0 ;
end
CONVERT Sample5 from MOLES to MOLES conserving MOLES
      GAMMA F7 Dum1 FILE GAM1
      ZERO 1.0 ;
end
CONVERT Sample6 from MOLES to MOLES conserving MOLES
      GAMMA F7 Dum1 FILE GAM1
      ZERO 1.0 ;

```

```
end
CONVERT Sample7 from MOLES to MOLES conserving MOLES
  GAMMA F7 Dum1 FILE GAM1
  ZERO 1.0 ;
end
CONVERT Sample8 from MOLES to MOLES conserving MOLES
  GAMMA F7 Dum1 FILE GAM1
  ZERO 1.0 ;
end
CONVERT Sample9 from MOLES to MOLES conserving MOLES
  GAMMA F7 Dum1 FILE GAM1
  ZERO 1.0 ;
end
CONVERT Sample10 from MOLES to MOLES conserving MOLES
  GAMMA F7 Dum1 FILE GAM1
  ZERO 1.0 ;
end

Mix "Sample1a" 1 Stream "Sample1"
Mix "Sample1a" 1 Stream "Sample2"
Mix "Sample1a" 1 Stream "Sample3"
Mix "Sample1a" 1 Stream "Sample4"
Mix "Sample1a" 1 Stream "Sample5"
Mix "Sample1a" 1 Stream "Sample6"
Mix "Sample1a" 1 Stream "Sample7"
Mix "Sample1a" 1 Stream "Sample8"
Mix "Sample1a" 1 Stream "Sample9"
Mix "Sample1a" 1 Stream "Sample10"
Write
clear
```

## Appendix C9

### Typical Gamma Fit usage for Multiple Samples

This appendix contains the complete data set for this example. This example illustrates a typical situation where multiple samples exist with different characterizations (i.e. number of components and their molecular weights). They are to be fit to the Gamma distribution model with varying parameters but with the same Eta (i.e. lower molecular weight).

This data set is also included in the accompanying CD-Rom.

**Streamz Driver File: InitialFitAllsamples.STZ**

TITLE 'Sample: 30/6-2 DST 1A Gas Samples GC-analysis; Report'  
 TITLE 'R005038'

GAMMAFILE GAM OPEN 'InitialFitAllSamples.gam'

CHAR 'DST30-6-2'  
 NAMES MW  
 FRC1 96  
 FRC2 107  
 FRC3 121  
 FRC4 134  
 FRC5 147  
 FRC6 161  
 FRC7 175  
 FRC8 190  
 FRC9 206  
 FRC10 222  
 FRC11 237  
 FRC12 264

Mix DST30-6-2 FRC1 MOLES  
 0.88 0.77 0.49 0.34 0.213 0.091 0.078 0.041 0.024  
 0.012 0.009 0.012

CHAR 'DST30-6-7'  
 NAMES MW  
 FRC1 92  
 FRC2 98  
 FRC3 107  
 FRC4 119  
 FRC5 133  
 FRC6 152  
 FRC7 170  
 FRC8 191  
 FRC9 225  
 FRC10 245  
 FRC11 259  
 FRC12 271  
 FRC13 305  
 FRC14 329  
 FRC15 360  
 FRC16 398  
 FRC17 431  
 FRC18 479  
 FRC19 542  
 FRC20 597  
 FRC21 620

Mix DST30-6-7 FRC1 Mass  
 2.08 4.05 3.97 4.13 3.82 4.57 4.42 6.29 4.48  
 5.55 4.77 5.59 4.84 4.42 4.26 4.04 3.57  
 4.01 2.69 3.76 11.57

CHAR '30-6-9'  
 NAMES MW  
 FRC1 94  
 FRC2 101  
 FRC3 106  
 FRC4 120  
 FRC5 136  
 FRC6 148

FRC7 166  
 FRC8 184  
 FRC9 201  
 FRC10 222  
 FRC11 243  
 FRC12 269  
 FRC13 294  
 FRC14 313  
 FRC15 352  
 FRC16 372  
 FRC17 421  
 FRC18 471  
 FRC19 532  
 FRC20 594  
 FRC21 620

Mix 30-6-9 FRC1 Mass

2.29	3.54	3.76	4.03	4.29	4.45	4.52	4.65	6.13
	5.19	4.65	4.41	5.24	4.35	3.69	4.91	4.06
	4.26	2.44	4.01	11.4				

CHAR '30-6-C14'

NAMES	MW
FRC1	91
FRC2	109
FRC3	116
FRC4	135
FRC5	147
FRC6	154
FRC7	167
FRC8	185
FRC9	196
FRC10	210
FRC11	222
FRC12	234
FRC13	247
FRC14	259
FRC15	267
FRC16	281
FRC17	295
FRC18	309
FRC19	539

Mix '30-6-C14' FRC1 Mass

3.16	5.16	4.34	4.34	3.61	2.96	4.46	3.86	4.12
	3.25	3.14	2.9	3	2.39	2	2.24	1.96
	2.13	36.14						

CHAR '30-6-4'

NAMES	MW
FRC1	90
FRC2	99
FRC3	106
FRC4	120
FRC5	139
FRC6	146
FRC7	160
FRC8	174
FRC9	194
FRC10	205
FRC11	218
FRC12	234

```
FRC13  248
FRC14  265
FRC15  470
```

```
Mix '30-6-4' FRC1 Mass
2.03  4.76  4.01  4.62  2.38  3.37  3.51  4.05  3.65
      3.93  3.04  3.42  3.15  3      45.87
```

```
CHAR DUM
COMP MW
DUM  100
```

```
STREAMFILE OUT:  OUTPUT 'streamz output/All_out.STR'
```

```
CONVERT 'DST30-6-2' from MOLES to MOLES conserving MOLES
  GAMMA FRC1 DUM
  FILE GAM
  ZERO 1.0 ;
CONVERT 'DST30-6-7' from MASS to MASS conserving MASS
  GAMMA FRC1 DUM
  FILE GAM
  ZERO 1.0 ;
CONVERT '30-6-9'      from MASS to MASS conserving MASS
  GAMMA FRC1 DUM
  FILE GAM
  ZERO 1.0 ;
CONVERT '30-6-C14'   from MASS to MASS conserving MASS
  GAMMA FRC1 DUM
  FILE GAM
  ZERO 1.0 ;
CONVERT '30-6-4'     from MASS to MASS conserving MASS
  GAMMA FRC1 DUM
  FILE GAM
  ZERO 1.0 ;
```

```
Mix "Sample1a" 1 Stream "DST30-6-2"
Mix "Sample1a" 1 Stream "DST30-6-7"
Mix "Sample1a" 1 Stream "30-6-9"
Mix "Sample1a" 1 Stream "30-6-C14"
Mix "Sample1a" 1 Stream "30-6-4(2)"
Write
clear
```

### Streamz Driver File: FitAllsamplesFixedEta.STZ

```
TITLE 'Sample: 30/6-2 DST 1A Gas Samples GC-analysis; Report'
TITLE 'R005038'
```

```
GAMMAFILE GAM OPEN 'FitAllSamplesFixedEta.gam'
```

```
CHAR 'DST30-6-2'
NAMES MW
FRC1  96
FRC2  107
FRC3  121
FRC4  134
FRC5  147
FRC6  161
FRC7  175
FRC8  190
FRC9  206
FRC10 222
FRC11 237
```

FRC12 264

Mix DST30-6-2 FRC1 MOLES

0.88	0.77	0.49	0.34	0.213	0.091	0.078	0.041	0.024
	0.012	0.009	0.012					

CHAR 'DST30-6-7'

NAMES	MW
FRC1	92
FRC2	98
FRC3	107
FRC4	119
FRC5	133
FRC6	152
FRC7	170
FRC8	191
FRC9	225
FRC10	245
FRC11	259
FRC12	271
FRC13	305
FRC14	329
FRC15	360
FRC16	398
FRC17	431
FRC18	479
FRC19	542
FRC20	597
FRC21	620

Mix DST30-6-7 FRC1 Mass

2.08	4.05	3.97	4.13	3.82	4.57	4.42	6.29	4.48
	5.55	4.77	5.59	4.84	4.42	4.26	4.04	3.57
	4.01	2.69	3.76	11.57				

CHAR '30-6-9'

NAMES	MW
FRC1	94
FRC2	101
FRC3	106
FRC4	120
FRC5	136
FRC6	148
FRC7	166
FRC8	184
FRC9	201
FRC10	222
FRC11	243
FRC12	269
FRC13	294
FRC14	313
FRC15	352
FRC16	372
FRC17	421
FRC18	471
FRC19	532
FRC20	594
FRC21	620

Mix 30-6-9 FRC1 Mass

2.29	3.54	3.76	4.03	4.29	4.45	4.52	4.65	6.13
	5.19	4.65	4.41	5.24	4.35	3.69	4.91	4.06
	4.26	2.44	4.01	11.4				

CHAR '30-6-C14'

NAMES MW

FRC1	91
FRC2	109
FRC3	116
FRC4	135
FRC5	147
FRC6	154
FRC7	167
FRC8	185
FRC9	196
FRC10	210
FRC11	222
FRC12	234
FRC13	247
FRC14	259
FRC15	267
FRC16	281
FRC17	295
FRC18	309
FRC19	539

Mix '30-6-C14' FRC1 Mass

3.16	5.16	4.34	4.34	3.61	2.96	4.46	3.86	4.12
	3.25	3.14	2.9	3	2.39	2	2.24	1.96
	2.13	36.14						

CHAR '30-6-4'

NAMES MW

FRC1	90
FRC2	99
FRC3	106
FRC4	120
FRC5	139
FRC6	146
FRC7	160
FRC8	174
FRC9	194
FRC10	205
FRC11	218
FRC12	234
FRC13	248
FRC14	265
FRC15	470

Mix '30-6-4' FRC1 Mass

2.03	4.76	4.01	4.62	2.38	3.37	3.51	4.05	3.65
	3.93	3.04	3.42	3.15	3	45.87		

CHAR DUM

COMP MW

DUM 100

STREAMFILE OUT: OUTPUT 'streamz output fix eta/All\_out.STR'

CONVERT 'DST30-6-2' from MOLES to MOLES conserving MOLES

GAMMA FRC1 DUM

FILE GAM

```
ZERO      1.0 ;
BOUND    91.0994
END
CONVERT  'DST30-6-7' from MASS to MASS conserving MASS
GAMMA FRC1 DUM IGNORE FRC21
FILE GAM
ZERO      1.0 ;
BOUND    91.0994
END
CONVERT  '30-6-9'      from MASS to MASS conserving MASS
GAMMA FRC1 DUM
FILE GAM
ZERO      1.0 ;
BOUND    91.0994
END
CONVERT  '30-6-C14'   from MASS to MASS conserving MASS
GAMMA FRC1 DUM
FILE GAM
ZERO      1.0 ;
BOUND    91.0994
END
CONVERT  '30-6-4'     from MASS to MASS conserving MASS
GAMMA FRC1 DUM
FILE GAM
ZERO      1.0 ;
BOUND    91.0994
END

Mix "Sample1a" 1 Stream "DST30-6-2"
Mix "Sample1a" 1 Stream "DST30-6-7"
Mix "Sample1a" 1 Stream "30-6-9"
Mix "Sample1a" 1 Stream "30-6-C14"
Mix "Sample1a" 1 Stream "30-6-4(2)"
Write
clear
```



## Appendix C10

### Gamma Distribution application to Reservoir Streams

This appendix contains a portion of the complete data sets for this example. Use of Gamma distribution is a viable alternative to convert the  $C_{n+}$  fractions in the reservoir stream to equivalent fractions in the process stream. This is done on a time-step basis for each grid-cell connection for maximum accuracy.

Various conversions are made to evaluate the alternative ways to use Gamma Distribution on a reservoir simulation results.

Small conversion files are included in full but only that portion of large conversion files are included that are relevant to the included portion of the stream file. Some comments may be altered/deleted in the portions of the data set included in this appendix to conserve space.

This complete data set included in the accompanying CD-Rom.

**Streamz Driver File: 2d-16c-tab.stz**

```

; Part 1: Generate tabulated 16-component stream file
include 16comp.chr
streamfile inp input 2d-16c.str
streamfile out output 2d-16c-tab.str

lump c6plus C6 C7+1 C7+2 C7+3 C7+4 C7+5 C7+6

filter valid c6plus moles gt 0.0
domain time t1 t2
tabulste time (days) and field and pres (bara) if valid

streamfile inp close
streamfile out close

; Part 2: Generate tabulated 9-component stream file
include 16comp.chr
streamfile inp input 2d-16c-tab.str

lump c6plus C6 C7+1 C7+2 C7+3 C7+4 C7+5 C7+6

include 9comp.chr
streamfile out output 2d-16c-tab-9c.str

convert 16comp from mole to mole
split N2      C1
split CO2     C2
split C1      C1
split C2      C2
split C3      C3
split IC4     C4
split C4      C4
split IC5     C5
split C5      C5
split C6      C6F3
split C7+1    C6F3
split C7+2    C6F3
split C7+3    C6F3
split C7+4    F4
split C7+5    F5F6
split C7+6    F5F6

filter valid c6plus moles gt 0.0
tabulste time (days) and field and pres (bara) if valid

eof

```

**Streamz Driver File: 2d-9c-to-16c.stz**

```

title 'Streamz run to convert 9c stream file to 16c based on'
title 'STR2CNV generated split factors'

include 9comp.chr

streamfile inp input 2d-16c-tab-9c.str
*streamfile inp input test.str

lump c6plus C6F3 F4 F5F6

include 16comp.chr

```

```
include 9c-to-16c.cnv
streamfile out output Results\2d-9c-16c.str
copy
```

**Streamz Driver File: gamma-9c-to-16c.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str

lump c6plus C6F3 F4 F5F6

include 16comp.chr

include gamma-sf-9c-16c.cnv

streamfile out output Results\gamma-9c-16c.str

copy

eof
```

**Streamz Driver File: gamma-16c-to-9c.stz**

```
title 'Streamz run to find Gamma Fit parameters'

include 16comp.chr

streamfile inp input 2d-16c-tab.str

gammafile gam1 open 16c.gam

include 9comp.chr
convert 16COMP from MOLES to MOLES conserving MASS warnings OFF
gamma C6 C6F3 file gam1
shape 1 .2 20
;origin 1.0
average 1.0

split N2      C1
split CO2     CO2
split C1      C1
split C2      C2
split C3      C3
split IC4     C4
split C4      C4
split IC5     C5
split C5      C5

streamfile out output 2d-16c-9c-gamma.str

copy
```

**Streamz Driver File: gamma-unrestrict-9-16.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str

lump c6plus C6F3 F4 F5F6

include 16comp.chr
```

```
include gamma-unrestrict-9-16.cnv
streamfile out output Results\gamma-unrestrict-9-16.str
copy
eof
```

**Streamz Driver File: gamma-fix\_ave\_zero-9-16.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str

lump c6plus C6F3 F4 F5F6

include 16comp.chr

include gamma-fix_ave_zero-9-16.cnv

streamfile out output Results\gamma-fix_ave_zero-9-16.str

copy

eof
```

**Streamz Driver File: gamma-fix\_init\_16\_params.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str

lump c6plus C6F3 F4 F5F6

include 16comp.chr

include gamma-fix_init_16_params.cnv

streamfile out output Results\gamma-fix_init_16_params.str

copy

eof
```

**Streamz Driver File: Fit\_16c.stz**

```
title 'Streamz run to find Gamma Fit parameters'

include 16comp.chr

streamfile inp input 2d-16c-tab.str

gammafile gam1 open Fit_16c.gam

char dum
comp mw
dum 220

convert 16COMP from MOLES to MOLES conserving MASS warnings OFF
gamma C6 dum file gam1
shape 1 .2 20
average 1.0

streamfile out output dum.str

copy
```

**Streamz Driver File: gamma-fix\_16\_params\_avg.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str

include 16comp.chr

include gamma-fix_16_params_avg.cnv

streamfile out output Results\gamma-fix_16_params_avg.str

copy

eof
```

**Streamz Driver File: gamma-vary\_16\_params.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str
lump c6plus C6F3 F4 F5F6

include 16comp.chr

include gamma-vary_16_params.cnv

streamfile out output Results\gamma-vary_16_params.str

copy

eof
```

**Streamz Driver File: gamma-9c-to-16c\_lmw.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str

lump c6plus C6F3 F4 F5F6

include 16comp_lmw.chr

include gamma-sf-9c-16c_lmw.cnv

streamfile out output Results\gamma-9c-16c_lmw.str

copy

eof
```

**Streamz Driver File: gamma-vary\_16\_params\_lmw.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str
lump c6plus C6F3 F4 F5F6

include 16comp_lmw.chr

include gamma-vary_16_params.cnv

streamfile out output Results\gamma-vary_16_params_lmw.str

copy
```

eof

**Streamz Driver File: gamma-fix\_init\_16\_params\_lmw.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str

lump c6plus C6F3 F4 F5F6

include 16comp_lmw.chr

include gamma-fix_init_16_params.cnv

streamfile out output Results\gamma-fix_init_16_params_lmw.str

copy

eof
```

**Streamz Driver File: gamma-vary\_16\_params\_lmw\_last.stz**

```
include 9comp.chr
streamfile inp input 2d-16c-tab-9c.str
lump c6plus C6F3 F4 F5F6

include 16comp_lmw_last.chr

include gamma-vary_16_params.cnv

streamfile out output Results\gamma-vary_16_params_lmw_last.str

copy

eof
```

**Characterization Include File: 9Comp.chr**

EOS 'PR'

CHAR '9COMP'

COMP	MW	TC	PC	AF	VS	AMOD	BMOD
		K	BARA				
C1	16.0924	190.115	45.9565	1.17E-02	-0.15913	1.00273	1.00249
CO2	44.01	304.206	73.8153	0.231	-8.20E-02	1	1.00005
C2	30.07	305.422	48.8011	9.08E-02	-0.113	1	1.00005
C3	44.097	369.822	42.4924	0.1454	-8.60E-02	1	1.00005
C4	58.124	420.326	37.5454	0.1879	-7.18E-02	0.985158	1.00262
C5	72.151	465.861	33.7388	0.24126	-4.79E-02	1.00905	0.988353
C6F3	103.717	571.351	28.8451	0.31566	3.21E-02	0.966199	0.979098
F4	165.409	685.364	21.3935	0.47973	0.11428	1	1.00005
F5F6	364.751	828.167	18.2002	0.71007	-0.37776	0.758841	0.92781

END

BIPS	C1	CO2	C2	C3	C4	C5	C6F3	F4	F5F6
C1	0	0.10425	7.00E-05	6.50E-04	6.80E-04	7.60E-04	2.77E-02	-1.04E-02	0.17952
CO2	0.10425	0	0.13	0.125	0.11642	0.115	0.115	0.115	0.115
C2	7.00E-05	0.13	0	0	0	0	0	0	0
C3	6.50E-04	0.125	0	0	0	0	0	0	0
C4	6.80E-04	0.11642	0	0	0	0	0	0	0
C5	7.60E-04	0.115	0	0	0	0	0	0	0
C6F3	2.77E-02	0.115	0	0	0	0	0	0	0
F4	-1.04E-02	0.115	0	0	0	0	0	0	0
F5F6	0.17952	0.115	0	0	0	0	0	0	0

**Characterization Include File: 16Comp.chr**

EOS 'PR'

```

CHAR '16comp'
COMP      MW           ~LMW       ~LMW ;AVERAGE
*
N2        28.0130
CO2       44.0100
C1        16.0430
C2        30.0700
C3        44.0970
IC4       58.1240
C4        58.1240
IC5       72.1510
C5        72.1510
C6        86.1780
C7+1     94.3919      88.69      88.9275
C7+2    106.239      100.17     99.8775
C7+3    121.201      113.13    113.04
C7+4    165.409      129.84    130.7975
C7+5    293.065      209.49    211.0475
C7+6    588.617      427.75    430.595
END

```

**Characterization Include File: 16Comp\_lmww.chr**

EOS 'PR'

```

CHAR '16comp'
COMP      MW           LMW       ~LMW ;AVERAGE
*
N2        28.0130
CO2       44.0100
C1        16.0430
C2        30.0700
C3        44.0970
IC4       58.1240
C4        58.1240
IC5       72.1510
C5        72.1510
C6        86.1780
C7+1     94.3919      88.69      88.9275
C7+2    106.239      100.17     99.8775
C7+3    121.201      113.13    113.04
C7+4    165.409      129.84    130.7975
C7+5    293.065      209.49    211.0475
C7+6    588.617      427.75    430.595
END

```

**Characterization Include File: 16Comp\_lmww\_last.chr**

EOS 'PR'

```

CHAR '16comp'
COMP      MW           LMW       ~LMW
*
N2        28.0130
CO2       44.0100
C1        16.0430
C2        30.0700
C3        44.0970
IC4       58.1240

```

C4	58.1240		
IC5	72.1510		
C5	72.1510		
C6	86.1780		
C7+1	94.3919	89.12	88.25
C7+2	106.239	99.61	96.63
C7+3	121.201	112.94	114.53
C7+4	165.409	131.58	140.97
C7+5	293.065	212.39	217.00
C7+6	588.617	433.79	406.15
END			

**Conversion Include File: 9c-to-16c.cnv**

Restore 16COMP

Convert 9COMP from MOLES to MOLES

```
Set C6PLUS MOLES/MOLE 0.386065788909694
Split C1          N2          0.00716733046958036
Split CO2         CO2          1
Split C1          C1          0.99283266953042
Split C2          C2          1
Split C3          C3          1
Split C4          IC4         0.281837288042795
Split C4          C4          0.718162711957205
Split C5          IC5         0.41619606621868
Split C5          C5          0.58380393378132
Split C6F3        C6          0.165511466343883
Split C6F3        C7+1       0.2962164187349
Split C6F3        C7+2       0.262330542599913
Split C6F3        C7+3       0.275941572321304
Split F4          C7+4        1
Split F5F6        C7+5       0.746636021432688
Split F5F6        C7+6       0.253363978567312
```

```
Set C6PLUS MOLES/MOLE 0.31264169621405
Split C1          N2          0.00799195762367191
Split CO2         CO2          1
Split C1          C1          0.992008042376328
Split C2          C2          1
Split C3          C3          1
Split C4          IC4         0.283950342246406
Split C4          C4          0.716049657753594
Split C5          IC5         0.417463909640725
Split C5          C5          0.582536090359275
Split C6F3        C6          0.168534825990815
Split C6F3        C7+1       0.296561267470653
Split C6F3        C7+2       0.261223370353691
Split C6F3        C7+3       0.273680536184841
Split F4          C7+4        1
Split F5F6        C7+5       0.746910715696768
Split F5F6        C7+6       0.253089284303232
```

**Conversion Include File: gamma-sf-9c-16c.cnv**

Restore 16COMP

gammafile gamfil open gamma-test.gam

```
Convert 9COMP from MOLES to MOLES, conserving mass, warnings OFF
gamma c6f3 c6 file gamfil
shape 1 0.2 20
origin 1.0
```

```
average 1.0
;bound 0.81
END
```

#### Conversion Include File: gamma-unrestrict-9-16.cnv

```
Restore 16COMP
gammafile gamfil open gamma-unrestrict.gam

Convert 9COMP from MOLES to MOLES, conserving mass, warnings OFF
gamma c6f3 c6 file gamfil
END
```

#### Conversion Include File: gamma-fix\_ave\_zero-9-16.cnv

```
Restore 16COMP
gammafile gamfil open gamma-unrestrict.gam

Convert 9COMP from MOLES to MOLES, conserving mass, warnings OFF
gamma c6f3 c6 file gamfil
ave 1
zero 1
END
```

#### Conversion Include File: gamma-fix\_init\_16\_params.cnv

```
Restore 16COMP
gammafile gamfil open gamma-unrestrict.gam

Convert 9COMP from MOLES to MOLES, conserving mass, warnings OFF

gamma c6f3 c6 file gamfil
; Parameters from fit of initial 16comp stream
Shape 0.75074
Average 1.00000
Bound 84.009
Origin 0.97621
END
```

#### Conversion Include File: gamma-fix\_16\_params\_avg.cnv

```
Restore 16COMP
gammafile gamfil open gamma-unrestrict.gam

Convert 9COMP from MOLES to MOLES, conserving mass, warnings OFF
gamma c6f3 c6 file gamfil
; Parameters from fit of initial 16comp stream (averages calculated
; in spreadsheet "2d-16c-tab.xls")
Shape 0.41828269
Average 1
Bound 84.7484883
Origin 0.964624708
END
```

#### Conversion Include File: gamma-vary\_16\_params.cnv

```
Restore 16COMP
gammafile gamfil open gamma-unrestrict.gam

Convert 9COMP from MOLES to MOLES, conserving mass, warnings OFF
gamma c6f3 c6 file gamfil
; Parameters from fit of initial 16comp stream (averages calculated
; in spreadsheet "2d-16c-tab.xls")

Set C6plus Moles/Mole 0.386065789
```

```
Shape 0.75074
Average 1
Bound 84.009
Origin 0.97621

Set C6plus Moles/Mole 00.247437086
Shape 0.7040513
Average 1
Bound 84.01158208
Origin 0.972730399

Set C6plus Moles/Mole 00.164958057
Shape 0.632318062
Average 1
Bound 83.96960979
Origin 0.964548901

Set C6plus Moles/Mole 00.082479029
Shape 0.412794382
Average 1
Bound 83.85288449
Origin 0.935260206

Set C6plus Moles/Mole 00.056149675
Shape 0.2
Average 1
Bound 83.791
Origin 0.90383

END
```

```

STREAMZ 1
Note 'Converted Streams from Streamz Files:'
Note '2d-16c-tab.stz'
Note '2d-16c-tab.str'
Char '9COMP'
Variable FIELD string
Variable GROUP string
Variable WELL string
Variable PVTNUM integer
Variable CONN_I integer
Variable CONN_J integer
Variable CONN_K integer
Variable TSTEP integer
Variable T1 time
Variable T2 time
Variable PFLAG integer
Variable PRES pressure
DATA

T1 (days) T2 (days) FIELD PRES (bar) Moles C1 MolesCO2 MolesC2 MolesC3 MolesC4 MolesC5 MolesC6F3 MolesF4 MolesF5F6
1086.3 1100 'FIELD' 282.497 2113.73 33.0679 295.464 226.855 160.728 117.846 624.567 526.391 702.665
1100 1130 'FIELD' 281.353 2122.89 33.1603 296.08 227.176 160.894 117.921 624.683 526.362 702.563
1130 1160 'FIELD' 280.212 2127.89 33.2108 296.417 227.351 160.984 117.961 624.747 526.346 702.507
1160 1190 'FIELD' 279.119 2131.85 33.2507 296.683 227.49 161.055 117.994 624.798 526.334 702.462
1190 1220 'FIELD' 278.035 2135.37 33.2862 296.92 227.613 161.119 118.023 624.842 526.323 702.424
1220 1250 'FIELD' 276.927 2138.65 33.3193 297.14 227.728 161.178 118.049 624.885 526.313 702.386
1250 1280 'FIELD' 275.941 2141.77 33.3507 297.35 227.838 161.235 118.075 624.925 526.303 702.351
1280 1310 'FIELD' 274.854 2144.75 33.3808 297.551 227.942 161.288 118.099 624.963 526.294 702.318
1310 1340 'FIELD' 273.843 2147.66 33.41 297.746 228.044 161.341 118.123 625.001 526.285 702.286

```

