1999

Automatic type curve matching for predicting gas wells production

Samson Tesfaslasie
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Tesfaslasie, Samson, "Automatic type curve matching for predicting gas wells production" (1999).
Graduate Theses, Dissertations, and Problem Reports. 979.
<https://researchrepository.wvu.edu/etd/979>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

**AUTOMATIC TYPE CURVE MATCHING FOR PREDICTING GAS WELLS
PRODUCTION**

Samson Tesfaslasie

A THESIS

**Submitted to the
College of Engineering and Mineral Resources
at
West Virginia University**

**In Partial Fulfillment of the Requirements for
The Degree of Master of Science in Petroleum and Natural Gas Engineering**

**Khashayar Aminian, Ph.D., Chair
Prof. Samuel Ameri, M.S
Ilkin Bilgesu, Ph.D
Dan Della-Giustine, Ph.D**

Department of Petroleum and Natural Gas Engineering

Morgantown, West Virginia

1999

**Keywords: Petroleum and natural gas engineering, Production forecasting of gas
well, Type curve matching, Aminian type curves, Decline curves, Gas in-
place, Non-darcy flow**

Copyright 1999 Samson Tesfaslasie

ABSTRACT

Automatic Type Curve Matching for Predicting Gas Wells Production

Samson Tesfaslasie

The determination of gas wells production using type curve matching is one of the methods used in evaluating gas reservoirs. Matching curves manually is time consuming and exposed to human error. Aminian type curves unlike the other type curves accounts the pressure dependency of gas properties and the effect of non-darcy factor. The sensitivity of Aminian type curves to non-darcy factor makes them hard to match with naked eyes.

Sfrac is a computer program generates type curves using conventional calculations for wells where all the reservoir properties are available. Using the idea of Sfrac, a program was developed to generate type curves with non-darcy factor and initial gas in-place in addition to the gas properties. This computer program which is know as MatchCurve automatically matches the production history to generated type curves. Based on the automatic match production of gas wells is predicted.

DEDICATION

This work is dedicated to my mom, Brikti Trfe, for being the greatest mother in her lifetime.

ACKNOWLEDGEMENT

The author wishes to express his appreciation to:

- Dr. K. Aminian for his direction and technical advice,
- Professor Sam Ameri for his moral support and being a father away from home,
- Dr. Ilkin Bilgesu and Dr. Dan Della-Giustina for their constructive criticism,
- My brothers, sisters, relatives and friends for their moral support, and
- My father, Tesfaslasie Gebregiorgis, for providing me the opportunity and for his moral support throughout my school years.

TABLE OF CONTENT

Content	Pages
List of Tables	vii
List of Figures	x
Nomenclature	xiii
CHAPTER I	
1. Introduction	1
CHAPTER II	
2. Literature Review	3
2.1 Volumetric Method	3
2.2 Material Balance	4
2.3 Reservoir Simulation	4
2.4 Decline Curve	5
2.4.1 Exponential Decline Curve	7
2.4.2 Hyperbolic Decline Curve	7
2.4.3 Harmonic Decline Curve	10
2.5 Type Curve	10
2.6 Type Curve Matching	24
2.7 Sfrac Computer Program	25
2.8 TypeCurveGenerator Program	27
CHAPTER III	
3. Methodology	31
3.1 CurveGenerator	31

3.2 Synchronization and Linear Interpolation	33
3.3 Sum of Least Squares Method	33
3.4 Iteration of G_i and F_{NDi}	37
3.5 Effect of Starting G_i and F_{NDi}	42
CHAPTER IV	
4. Results	46
4.1 Matching Results	46
4.2 Results On Predicting Gas Production	49
4.2.1 Sample One	49
4.2.2 Sample Two	54
CHAPTER V	
5. Discussions	59
5.1 G_i and F_{NDi} in curve matching	59
5.2 Prediction of gas wells production	60
5.3 Sensitivity of Type curves	61
5.4 Brief discussion about MatchCurve	62
CHAPTER VI	
6. Conclusions	64
CHAPTER VII	
7. Recommendation	66
Reference	67
Appendix	
A. Sample Runs	69

B. List of Sfrac	87
C. List of MatchCurve	102

LIST OF TABLES

Tables	Pages
Table 3.1 Verification of the predicted data with the actual data	54
Table 3.5 Run of different Starting F_{NDi} for actual F_{NDi} of 1.54 and G_i of 1.9 Bcf	38
Table 4.1 Match verification	47
Table 4.2 G_i and F_{NDi} of actual and matched Curves	48
Table 4.3 Gas prediction of sample one	50
Table 4.4 G_i and F_{NDi} results of sample one	50
Table 4.5 Gas prediction of sample two	55
Table 4.6 G_i and F_{NDi} results of sample one	55
Table 5.1 Sample runs in matching	60
Table 5.2 Correlation of Permeability and F_{NDi}	60
Table 5.3 Sample runs in predicting gas production	61
Table A.1 Actual data ($F_{NDi}=1.17$ and $G_i = 1.9$ Bcf) and best match	69
Table A.2 MatchCurve results for actual data ($F_{NDi} = 1.17$ and $G_i = 1.9$ Bcf)	69
Table A.3 Actual data ($F_{NDi}=1.25$ and $G_i = 13.8$ Bcf) and best match	70
Table A.4 MatchCurve results for actual data ($F_{NDi}=1.25$ and $G_i = 13.8$ Bcf)	70
Table A.5 Actual data ($F_{NDi}=1.46$ and $G_i = 2.665$ Bcf) and best match	71
Table A.6 MatchCurve results for actual data ($F_{NDi}=1.46$ and $G_i = 2.665$ Bcf)	71
Table A.7 Actual data ($F_{NDi}=1.55$ and $G_i = 40.49$ Bcf) and best match	72
Table A.8 MatchCurve results for actual data ($F_{NDi}=1.55$ and $G_i = 40.49$ Bcf)	72
Table A.9 Actual data ($F_{NDi}=1.57$ and $G_i = 24.9$ Bcf) and best match	73
Table A.10 MatchCurve results for actual data ($F_{NDi}=1.57$ and $G_i = 24.9$ Bcf)	73
Table A.11 Actual data ($F_{NDi}=1.58$ and $G_i = 18.168$ Bcf) and best match	74

Table A.12 MatchCurve results for actual data ($F_{NDi}=1.58$ and $G_i = 18.168$ Bcf)	74
Table A.13 Actual data ($F_{NDi}=1.81$ and $G_i = 5.29$ Bcf) and best match	75
Table A.14 MatchCurve results for actual data ($F_{NDi}=1.81$ and $G_i = 5.29$ Bcf)	75
Table A.15 Actual data ($F_{NDi}=2.2$ and $G_i = 1.77$ Bcf) and best match	76
Table A.16 MatchCurve results for actual data ($F_{NDi}=2.2$ and $G_i = 1.77$ Bcf)	76
Table A.17 Actual data ($F_{NDi}=3.02$ and $G_i = 12.8$ Bcf) and best match	77
Table A.18 MatchCurve results for actual data ($F_{NDi}=3.02$ and $G_i = 12.8$ Bcf)	77
Table A.19 Actual data ($F_{NDi}=4.76$ and $G_i = 1.29$ Bcf) and best match	78
Table A.20 MatchCurve results for actual data ($F_{NDi}=4.76$ and $G_i = 1.29$ Bcf)	78
Table A.21 Actual data (1453 days) and predicting curve (1444 days)	79
Table A.22 Results of matched type curve	79
Table A.23 Actual data (1115 days) and predicting curve (1444 days)	80
Table A.24 Results for predicting for 329 days	80
Table A.25 Actual data (802 days) and predicting curve (1440 days)	81
Table A.26 Results for predicting for 638 days	81
Table A.27 Actual data (504 days) and predicting curve (1417 days)	82
Table A.28 Results for predicting for 913 days	82
Table A.29 Actual data (207 days) and predicting curve (943 days)	83
Table A.30 Results for predicting for 736 days	83
Table A.31 Match verification	84

LIST OF FIGURES

Figures	Pages
Figure 2.1 Decline of flow rate - Linear plot	6
Figure 2.2 Decline of flow rate - Semi-log plot	8
Figure 2.3 Extrapolation of hyperbolic decline - Log-log plot	11
Figure 2.4 Hyperbolic curve matching method	12
Figure 2.5 Flow rate versus time of exponential, hyperbolic, and harmonic decline	13
Figure 2.6 Type curve for Arps empirical rate-time decline unit solution ($D_i = 1$)	14
Figure 2.7 Dimensionless flow rate versus dimensionless time: type curve for transient and pseudo-steady state flow	17
Figure 2.8 Fetkovich's decline type curve for gas wells	18
Figure 2.9 The effect of non-darcy flow on Aminian's type curves	22
Figure 2.10 The effect of X_i on Aminian's type curves	23
Figure 2.11 Example of type curve matching process	26
Figure 2.12 Gas properties calculation form of Sfrac	28
Figure 2.13 Decline curve analysis form of Sfrac	29
Figure 2.14 Type curve and result form of Sfrac	29
Figure 2.15 Input and output form of TypeCurveGenerator	30
Figure 3.1 Properties calculation form for CurveGenerator	34
Figure 3.2 Input and output of CurveGenerator form	35
Figure 3.3 Comparison of Sfrac and CurveGenerator decline curves	36

Figure 3.4 Plot of generated curves for different values of G_i and F_{NDi}	38
Figure 3.5 Plot of generated curves for different values of G_i	39
Figure 3.6 Plot of generated curves for different values of F_{NDi}	40
Figure 3.7 Simple flow chart of generating and comparing process	41
Figure 3.8 Error versus Starting F_{NDi} (Actual $G_i = 1.9$ Bcf & $F_{NDi} = 1.54$)	43
Figure 3.9 Gas properties calculation Form for MatchCurve Program	44
Figure 3.10 Result form for MatchCurve computer program	45
Figure 4.1 Actual ($F_{NDi} = 2$) and matched curves	48
Figure 4.2 Prediction of gas production for 2 years	51
Figure 4.3 Actual data (1.4 years) and predicting curve (3.4 years)	52
Figure 4.4 Total actual data and predicting curve of sample one	53
Figure 4.5 Prediction of gas production for 3.2 years	56
Figure 4.6 Actual data (2.25 years) and predicting curve (5.4 years)	57
Figure 4.7 Total actual data and predicting curve of sample two	58
Figure 5.1 Predicting production with small data	63
Figure A.1 Visual verification of match for actual data ($F_{NDi} = 1.17$ and $G_i = 1.9$)	69
Figure A.2 Visual verification of match for actual data ($F_{NDi} = 1.25$ and $G_i = 13.8$ Bcf)	70
Figure A.3 Visual verification of match for actual data ($F_{NDi} = 1.46$ and $G_i = 2.665$ Bcf)	71
Figure A.4 Visual verification of match for actual data ($F_{NDi} = 1.55$ and $G_i = 40.49$ Bcf)	72

Figure A.5 Visual verification of match for actual data	
($F_{NDi}=1.57$ and $G_i = 24.9$ Bcf)	73
Figure A.6 Visual verification of match for actual data	
($F_{NDi}=1.58$ and $G_i = 18.168$ Bcf)	74
Figure A.7 Visual verification of match for actual data	
($F_{NDi}=1.81$ and $G_i = 5.29$ Bcf)	75
Figure A.8 Visual verification of match for actual data	
($F_{NDi}=2.2$ and $G_i = 1.77$ Bcf)	76
Figure A.9 Visual verification of match for actual data	
($F_{NDi}=3.02$ and $G_i = 12.8$ Bcf)	77
Figure A10 Visual verification of match for actual data	
($F_{NDi}=4.76$ and $G_i = 1.29$ Bcf)	78
Figure A.11 Plot of production history and type curve	79
Figure A.12 Plot of predicting curve (329 days of prediction) and history data	80
Figure A.13 Plot of predicting curve (638 days of prediction) and history data	81
Figure A.14 Plot of predicting curve (736 days of prediction) and history data	82
Figure A.15 Plot of predicting curve (736 days of prediction) and history data	83

NOMENCLATURE

A	Area, ft ²
a	Coefficient of deliverability equation, psi ² /(cp)(Mcf/D)
b	Coefficient of deliverability equation, psi ² /(cp)(Mcf/D)
b ₁	hyperbolic exponent, dimensionless
B	Turbulence factor, ft ⁻¹
B _g	Gas Formation volume factor, bbl/scf
B _{gi}	Initial gas formation volume factor, bbl/scf
C	Constant, dimensionless
C ₁	Correlation Coefficient, defined by equation (28)
C _g	Gas compressibility, psia ⁻¹
C _{gi}	Initial gas compressibility, psia ⁻¹
C _t	Total compressibility, psia ⁻¹
D	Decline rate, day ⁻¹
D _i	Initial decline rate, day ⁻¹
F _B	Turbulence factor, (Mcf/D) ⁻¹
F _{NDi}	Non-Darcy Factor, defined by equation (25)
F _{taD}	Pseudo-time ratio, defined by equation (21)
G _D	Dimensionless cumulative gas production, dimensionless
G _i	Initial gas reserve, Bcf
G _p	Cumulative Gas production, Mcf/D
h	Length of formation, ft
k	Permeability, md

L	Drawdown Constant, defined by equation (24)
n	Exponent of back pressure equation, dimensionless
N _{FO}	Forchheimer Number, defined by equation (29)
ΔP	Difference of pressure, psia
P _D	Dimensionless pressure, dimensionless
P _i	Initial Pressure, psia
P _P	Pseudo-Pressure, psia ² / cp
ΔP_P	Difference of Pseudo-Pressures, psia ² / cp
P _R	Reservoir pressure, psia
P _{WF}	Bottom hole pressure, psia
Δq	Difference in flow rate, Mcf/D
q	Flow rate, Mcf/D
q _i	Initial flow rate, Mcf/D
q _D	Dimensionless flow rate, dimensionless
qDd	Fetkovich's dimensionless flow rate, dimensionless
q ₁	Flow rate, Mcf/D
q ₂	Flow rate, Mcf/D
r _e	Reservoir diameter, ft
r _w	Wellbore hole diameter, ft
S	Skin factor, dimensionless
S _{CA}	Shape-skin factor, dimensionless
S _g	Gas saturation, fraction
T	Temperature, °R

t	Time, day
Δt	Difference in time, day
t_1	Time, day
t_2	Time, day
t_D	Dimensionless time, dimensionless
t_{Dd}	Fetkovich's dimensionless time, dimensionless
X_i	BackPressure ratio, dimensionless
<i>Volume</i>	Volume of gas, ft ³
Z	Gas deviation factor, dimensionless
Z_i	Initial gas deviation factor, dimensionless
Z_{WF}	Gas deviation factor of flowing bottom hole pressure, dimensionless

Greek Symbols

β	Turbulence factor ft ⁻¹
μ	Viscosity, cp
μ_{gi}	Initial gas viscosity, cp
μ_g	Gas viscosity, cp
ϕ	Porosity, fraction

CHAPTER I

INTRODUCTION

In the petroleum and natural gas industry, predicting gas wells production is one of the main tasks of reservoir engineers. Reservoir engineers use material balance equations, reservoir simulation and type curve analysis as the main tools to predict the performance of gas wells. The objective of reservoir engineers is to analyze the past performance, forecast future production and estimate oil and gas reserve of wells.

Reservoir engineers ask themselves, what are the interests of gas investors? Like all investors in any industry, gas investors are interested in knowing the risk and the reward of their investment. Operators want to know the amount of gas reserve and the rate of production decline so that they can be able to sign contracts to sell the gas. The more knowledge the Operators have about the future production of their wells the better business decisions they can make to increase the rate of return for the gas investors.

To keep the investors interest, reservoir engineers use all their knowledge to understand the behavior of reservoirs and the effect of reservoir properties on the production rate. So far, reservoir engineers have done many studies to determine various reservoir properties in order to forecast future production of gas wells.

Type curve matching is one of the commonly used techniques for determining reservoir properties and predicting gas well production. This shape matching is done by plotting log-log production history on a transparent paper and try to fit it to log-log type curve plots. The two log-log plots do not exactly fit each other all the time therefore it is difficult to determine which type curve fits the production history curve. It is very a

tidous processes for many reservoir engineers and at the same time, it is highly exposed to human error.

The main objective of this study is to avoid the use of manual matching process and to develop an automatic type curve matching process using computer program to forecast gas wells production. The advantage of using a computer program to match production history with type curve is to save many hours of hard work and to obtain accurate matches.

CHAPTER II

LITERATURE REVIEW

Reservoir engineers study gas reservoir parameters to estimate gas reserve and total gas production and to predict gas production rate and life of gas wells. To study the gas reservoir properties, what kind of information is available for engineers? After a well is drilled, the first thing engineers encounter is reservoir pressure. The difference of pressure between the reservoir pressure and the surface pressure causes the gas to flow to the surface. The initial flow rate is high and then the flow rate starts to decline with time. How fast will the production rate decline? How long will the production lasts? How much gas in-place does the reservoir have? How much of the resource is recoverable? Is it economical to use artificial lift to increase recovery? These are some of the questions that reservoirs engineer face on daily bases. Volumetric equations, material balance equations, reservoir simulation and type curve analysis are methods used to determine reservoir parameters and to analyze gas production.

2.1 Volumetric Method

This method is based on the geological maps, usually obtained from well logs and core data.¹ Volumetric equations, as the name indicates, are used to calculate the volume of oil and gas in a reservoir using structural and isopach maps. The saturation, formation volume factor and porosity are the variables used to calculate the initial resources. Trapezoidal rule and Simpson's rule are the most commonly used methods for calculating the bulk volume.¹

2.2 Material Balance

Material balance equations are developed based on conservation of mass. The assumption made in developing this equation is that the reservoir is treated as a container with uniform pressure throughout. If the reservoir pressure is precisely averaged during production, material balance equation can accurately predict the production figures.¹ Material balance can be applied to all types of reservoirs such as solution-gas-drive, gas-cap-drive and water-drive-reservoirs.

2.3 Reservoir Simulation

Reservoir simulation is another type of approach to estimate performance of a reservoir. Simulate means literally “to assume the appearance without the reality.”² The method assumes the shape of the reservoir using geological data such as structural and isopach maps. Reservoir simulators deal with fluid flow through porous media and determine how the pressure drops in the entire reservoir. Fluid flow through porous media is caused by fluid expansion, displacement, gravity drainage and capillary imbibition.² Using a reservoir simulation model, one can see how the reservoir is going to behave to different operating conditions. Optimum production conditions and better recovery can be achieved by using simulator.

From 1930 to 1960, reservoir simulation has been done using analytical approach, later the numerical method followed.² Analytical solution is an exact and continuous solution for an approximate problem. Numerical model uses a finite difference technique to solve an exact problem. The partial differential equation (PDE) of fluid through porous media is approximated to finite difference. The solution obtained from the numerical model is an approximate and discrete solution.

2.4 Decline Curve Analysis

The technique of estimating future production and reserve by plotting production rate versus time is called Decline-Curve analysis.³ Decline-Curve analysis assumes that the producing conditions and reservoir parameters throughout the life of the reservoir are unchanged.⁴ The decline curve analysis began as a plot of production rate versus time, in a Cartesian plot. Then engineers tried to fit the plotted points on a straight line to extrapolate the future production and reserve as shown in Figure 2.1. From this plot, a straight line could be achieved at the latest time since the decline rate is not constant.³ In order to get a better straight-line, flow rate versus time data were plotted on semi-log scaled plot. This semi-log plot is the commonly used decline curve to extrapolated future production³ as shown in Figure 2.2. However, all wells do not achieve a straight line in the semi-log scaled graph then this leads to an investigation of another plot which is know as Type-Curve analysis.³ Production decline rate is defined as:

$$D_1 = \frac{\left(\frac{\Delta q}{q_i} \right)}{\Delta t} \quad (1)$$

The decline rate is inversely proportional to time as defined in equation (1). The use of semi-log scale came into effect to keep the decline rate constant as shown in equation (2) and Figure 2.2.

Fig. 2.1 Decline of flow rate - Linear plot³

$$D = \frac{(\text{Log}(\Delta q))}{\Delta t} \quad (2)$$

The three types of decline-curves are exponential, hyperbolic and harmonic. Exponential and hyperbolic decline curves are commonly used.^{1, 3, 5}

2.4.1 Exponential Decline-Curve

Exponential decline curve is also known as “Constant-Percentage Decline.” It is the most commonly used decline-curve because of its simplicity.³ Using equation (2) the exponential term is derived as shown below.

$$\log(\Delta q) = \log(q) - \log(q_i) \quad (3)$$

$$\log(\Delta q) = \log\left(\frac{q}{q_i}\right) \quad (4)$$

$$D = -\frac{\left(\frac{\log(\Delta q)}{q_i}\right)}{\Delta t} \quad (5)$$

$$q = q_i \cdot e^{-Dt} \quad (6)$$

2.4.2 Hyperbolic Decline-Curve

Hyperbolic decline curves are curves where the decline rate is no longer constant. Comparing to exponential decline-curve, hyperbolic decline-curve equations estimate for a longer production time of a well.¹

Fig. 2.2 Decline of flow rate - Semi-log plot. ³

When the exponential decline-curve analysis is not applicable, it is advisable to use hyperbolic decline-curve to estimate the reserve and future production rate.³ What makes hyperbolic decline curves more complex, is that the decline rate is no longer a constant. Hyperbolic curves have hyperbolic exponent, b_1 , between 0 and 1. Exponential decline curves have hyperbolic exponent of 0 while harmonic decline curves have hyperbolic exponent of 1. Hyperbolic decline-curve is derived as shown below by substituting equation (1) into equation (7).

$$\frac{D}{D_i} = \left(\frac{q}{q_i} \right)^{b_1} \quad (7)$$

$$\frac{\left(\frac{\Delta q}{q \cdot \Delta t} \right)}{D_i} = \left(\frac{q}{q_i} \right)^{b_1} \quad (8)$$

$$q = q_i (1 + b \cdot D_i \cdot t)^{\frac{-1}{b_1}} \quad (9)$$

In equation (9), t is the time taken to reach the flow rate from q_i to q . The hyperbolic exponent, b_1 , is determined from past rate-time data. The difficult part of this method is determining b_1 . After b_1 is determined, flow rate can be calculated at any point of time.

Log-log plot of a hyperbolic decline curve is also another way of analyzing decline curves. This method involves trial and error until the log-log plot of flow rate versus time is a straight line as shown in Figure 2.3. Where C is the constant number that makes the log-log plot a straight line. Once the straight line is obtained, b_1 is calculated

from the slope of the line and then the production rate is calculated from C and b_1 .³ Another way of determining the reservoir properties and predicting a gas well production is by plotting the production history in a semi-log graph and trying to match it to the hyperbolic decline curves as shown in Figure 2.4.

2.4.3 Harmonic Decline Curve

Harmonic decline-curve is the third type of decline curve analysis that is rarely used. It is usually used for wells where production is affected by "gravity drainage" and for wells with long life. As mentioned in the previous section, the hyperbolic exponent of this decline-curve is equal to 1.³ Harmonic decline curves are defined as:

$$q = q_i \cdot (1 + D_i \cdot t)^{-1} \quad (10)$$

The difference and similarity of the three types of decline curves are shown in Figure 2.5 on the Cartesian plot.¹ Arps curves, which are shown on Figure 2.6, are also used for any of the three types of decline curves by finding a best matching type curve to a production history of a well.⁶

2.5 Type Curve Analysis

A type curve is mostly defined as a log-log plot of dimensionless flow rate or pressure verses dimensionless time. At "constant pressure test" decline of flow rate was observed. While at "constant rate test" decline of pressure was observed with time.⁷ Pressure derivative is another type curve used in calculating reservoir properties and

Fig. 2.3 Extrapolation of Hyperbolic decline-Log-Log plot.³

Fig.2.4 Hyperbolic curve matching method. ³

Fig. 2.5 Flow rate versus Time of Exponential, Hyperbolic, and Harmonic Decline. ¹

Fig. 2.6 Type curves for Arps empirical rate-time decline - for $D_i = 1$.⁸

Forecasting future gas production. From Arp's curves the dimensionless flow rate (q_D) and dimensionless time (t_D) are defined as: ⁶

$$q_D = \frac{q(t)}{q_i} \quad (11)$$

$$t_D = D_i \cdot t \quad (12)$$

The analytical solution of the dimensionless flow rate and time is defined as: ⁶

$$q_D = \frac{141.2 \cdot q(t) \cdot \mu_g \cdot B_g}{K \cdot h \cdot (P_i - P_{wf})} \quad (13)$$

$$t_D = \frac{0.00634 \cdot K \cdot t}{\phi \cdot \mu_g \cdot C_t \cdot r_w^2} \quad (14)$$

The analytical solution of the dimensionless pressure is defined as: ⁹

$$P_D = \frac{k \cdot h \cdot \Delta P}{141.2 \cdot q \cdot B_g \cdot \mu_g} \quad (15)$$

In 1973, Fetkovich ⁶ combined the "analytical" constant-pressure solution with Arps "empirical" decline curves and generated a set of dimensionless type curves for oil wells as shown in Figure 2.7. Later in 1980, he developed another set of type curves for gas wells using an "empirical back pressure" (16) and material balance equation (17). These type curves are shown in Figure 2.8.

$$q = C \cdot (P_R^2 - P_{WF}^2)^n \quad (16)$$

$$P_R = -\left(\frac{P_i}{G_i}\right) \cdot G_P + P_i \quad (17)$$

Where,

$$q_D = \frac{q}{q_i} \quad (18)$$

$$t_D = \frac{q \cdot t}{G_i} \quad (19)$$

Fetkovich's gas type curves do not account for pressure dependency of gas properties and non-darcy effect caused by turbulence flow. Carter¹⁰ and Frain & Wattenbarger¹¹ by realizing the change of gas properties with pressure developed separate sets of type curves to overcome this problem. However, they neglected the effect of the non-darcy flow, which is caused by the high velocity of gas at the sandface.

In 1986, Aminian¹² and Schmidh¹³ separately developed type curves that accounted both for the change of gas properties with pressure and the non-darcy effect. It was also found out that these type curves are affected by the initial reservoir pressure, permeability and skin factor. As mentioned in previous sections, the decline curve and type curves assume that the reservoir properties do not change for the entire production time of a well. In 1990, Aminian¹⁴ challenged this idea by providing some facts that it is impossible for the assumption to hold for the entire life of the reservoir. Well stimulation

Fig. 2.7 Dimensionless flow rate versus Dimensionless time: Type curve for transient and Pseudo-steady state flow. ⁸

Fig. 2.8 Fetkovich's decline type curve for gas wells. ⁷

or damage, shape of drainage area, and backpressure are some of the reservoir properties that can be altered during production time. Aminian et al,¹⁵ used pseudo-pressure, pseudo-time and non-darcy factor to implement the pressure dependency of gas properties, the non-darcy effect and the change of reservoir properties during production into the type curves. Aminian's general equation generates type curves considering all the necessary reservoir parameter changes during production. The general equation is defined as:

$$\ln(q_D) + 2 \cdot (1 - F_{NDi}) \cdot (q_D - 1) + \frac{(F_{NDi} \cdot F_{taD} \cdot t_D)}{L \cdot \left(1 - \frac{1}{X_i}\right)} = 0 \quad (20)$$

F_{taD} is a pseudo-time that accounts for the pressure dependency of gas property and defined as:¹⁵

$$F_{taD} = \frac{\left[\int_0^t \frac{1}{\mu_g \cdot C_g} dt \right]}{\frac{t}{\mu_{gi} \cdot C_{gi}}} \quad (21)$$

Pseudo-pressure and X_i , which is the ratio of reservoir pressure and bottom hole pressure, are defined as:¹⁵

$$P_p(P) = \int_0^P \frac{2 \cdot P}{\mu_g \cdot Z} dp \quad (22)$$

$$X_i = \frac{\left(\frac{P_i}{Z_i} \right)}{\left(\frac{P_{wf}}{Z_{wf}} \right)} \quad (23)$$

L is a term that accounts the pressure dependency of gas property is defined as: ¹⁵

$$L = [P_p(P) - P_p(P_{wf})] \cdot \frac{\mu_{gi} \cdot C_{gi}}{2 \cdot X_i} \quad (24)$$

The non-darcy effect (F_{NDi}) is defined as: ¹⁵

$$F_{NDi} = 1 + \frac{b \cdot q_i}{a} \quad (25)$$

a, the coefficient of the deliverability equation, is defined as: ¹⁵

$$a = 1422 \cdot \frac{\left(\ln \left(\frac{r_e}{r_w} \right) - 0.75 + S_{CA} + S \right)}{K \cdot h} \quad (26)$$

b, the other coefficient of the deliverability equation, is defined as: ¹⁵

$$b = \left[\frac{1422 \cdot T}{K \cdot h} \right] \cdot C_1 \cdot F_B \quad (27)$$

C_1 , N_{FO} and F_B are defined as: ¹⁵

$$C_1 = \left[1 - \left(\frac{r_e}{r_w} \right) \cdot \left(\frac{\mu_{gi}}{\mu_g} \right) \cdot (N_{FO})^{-0.028} \right] \quad (28)$$

$$N_{FO} = \frac{2.224 \times 10^{-12} \cdot B \cdot \mu_g \cdot k \cdot q}{h \cdot \mu_{gi} \cdot r_w} \quad (29)$$

$$F_B = \frac{(2.224 \times 10^{-15} \cdot B \cdot \mu_g)}{h \cdot \mu_{gi} \cdot r_w} \quad (30)$$

X_i and L are constant as long as the backpressure remains constant. Both properties depend on the initial reservoir pressure, bottom hole pressure and the gas composition or specific gas gravity. F_{tD} , which is pressure dependent, varies with pressure. Aminian's general equation included the standard gas deviation factor (Z-factor) into the material balance equation and the dimensionless values of X_i . As pressure varies the standard gas deviation factor varies accordingly, therefore the implementation of standard gas deviation factor into X_i (23), and material balance equation (31) accounts for the change of gas property caused by pressure variation.¹⁵ Aminian type curves are shown on Figure 2.9 and 2.10 for different X_i and F_{NDi} values respectively. The solution of the general equation is applicable for pseudo-radial flow and fractured wells that undergo pseudo-radial flow.^x

$$G_P = \frac{\left(\frac{P_i}{Z_i} - \frac{P}{Z} \right)}{\frac{P_i}{Z_i}} \cdot G_i \quad (31)$$

Two methods had been utilized to generate Aminian's type curves. The first method is solving t_D and q_D using material balance equation and deliverability equation

Fig. 2.9 The effect of non-darcy flow on Aminian's type curves. ¹⁵

Fig. 2.10 The effect of X_i on Aminian's type curves.¹⁵

for every pressure increment. The second method is polynomial regression, which used by Abidi¹⁷ for F_{taD} as a function of t_D .

From the general equation, interesting observation were also made by Aminian. If F_{taD} , L , and F_{NDi} are equal to one then the general equation is simplified to the simplest form of decline curve that is exponential decline curve. Ignoring non-darcy effect that is by keeping F_{NDi} equal to 1 the other observation about the general equation was made. The type curves generated with this condition resemble the type curves generated by Farim and Watterbarges.^{15, 16} These two observations show the consistency of the general equation. This general equation is valid for single-phase flow and pseudo-steady state flow regime.¹⁵

2.6 Type Curve Matching

It is essential to know X_i and F_{NDi} to generate type curves. X_i affects the position of the curve as shown in Figure 2.10, while F_{NDi} changes the shape of the curve as shown on Figure 2.9. Since X_i is available from the producing well, F_{NDi} was iterated for different values and different type curves were generated. Then by superimposing the production history of a well on the top of the generated curves, a match was made and as the result, F_{NDi} read from the matched curve. Once a match is found, q_i and G_i were calculated using equations (32) and (33).¹⁸ The deliverability coefficient (a and b) were determined using equation (34) and (35). An example of matching process is shown in Figure 2.11.

$$q_i = \frac{q}{q_D} \quad (32)$$

$$G_i = \frac{q_i \cdot t}{t_D} \quad (33)$$

$$a = \frac{[P_p(P_i) - P_p(P_{wf})]}{q_i \cdot F_{NDi}} \quad (34)$$

$$b = \frac{(F_{NDi} - 1) \cdot a}{q_i} \quad (35)$$

2.7 Sfrac Computer Program

Sfrac is a simple analytic gas model that generates production decline curves of a well. The program has two parts; the first part of the program calculates the gas properties and pseudo-pressure for any pressure increment. The second part generates decline production curves and dimensionless type curves. It also prints out F_{NDi} , a , b , and G_i as part of the output.

For the first part of the program, the input parameters are the initial reservoir pressure, reservoir temperature and gas gravity or gas components. For the second part of the program, porosity, permeability, skin factor, shape factor, bottom hole flowing pressure, wellbore diameter, gas saturation, and area of the reservoir are the input parameters.

Pseudo-pressure was calculated using equation (22). The coefficient of the deliverability equations (a and b) were calculated using the equations (26) and (27)

Fig. 2.11 Sample of matching process.¹⁸

respectively. The initial gas in-place was calculated using volumetric equations as shown in equation (40). Flow rate for each pressure increment was calculated using deliverability and material balance equations then the time of production was calculated using equation (36) follows.

$$\Delta t = \frac{\Delta G_p}{q} \quad (36)$$

Equation (18) and (19) are used to calculate the dimensionless flow rate and dimensionless time respectively. Dimensionless production is also obtained using equation (37).

$$G_D = \frac{G_p}{G_i} \quad (37)$$

Bottom hole flowing pressure, skin factor and shape factor can be changed any time during production and the program accounts the changes made in the calculation. In 1992, Abidi underlines the limitation of the program as follows: the reservoir has to be homogenous, the reservoir has to be finite and also in pseudo-steady state, the reservoir has to be a gas reservoir with no oil or water production involved, and production should not be supported by a water drive.¹⁷ List of Sfrac computer program is shown in Appendix B. Figure 2.12 and Figure 2.13 show the forms of Sfrac of a sample run.

2.8 Type Curve Generator

Abidi¹⁷ has worked to develop a direct method of generating type curves using polynomial regression by creating a correlation with the Aminian's general equation (26).

Fig. 2.12 Gas properties calculation form.

Fig. 2.13 Decline curve analysis form.

Fig. 2.14 Type curve and Result form

The program was written in Fortran and it is updated to Visual Basic 5.0. The Second and third degree order polynomial equations were used depending on initial reservoir pressure. Third degree order polynomial equations were used for initial reservoir pressure ranging between 500 to 2600 psia. Second degree order polynomial equations were used for initial reservoir pressure ranging from 2600 to 5000 psia. Abidi determined the coefficients of the polynomial equation.¹⁷

The input properties of the program are P_i , X_i , and F_{NDi} and the outcome of the program is a dimensionless type curve or a list of t_D and q_D values. A sample run is shown in Figure 2.15.

The screenshot shows a software window titled "The TypeCurveGenerator". At the top right, there is a button labeled "Save The Change for later retrieval". The main area is divided into two sections. The left section contains input fields for "Fndi" (value 2), "Xi" (value 5.4), and "Pi (psi)" (value 3000). Below these are two buttons, "Run" and "Exit". There is a checkbox labeled "Read Data From File" which is checked. Below the checkbox is a file selection button with a folder icon and a text box containing "a:" and "Td.txt". The right section contains a table with two columns, "Td" and "Qd", and 13 rows of data.

	Td	Qd
1	0.001000	0.9825
2	0.002000	0.9812
3	0.003000	0.9800
4	0.004000	0.9788
5	0.005000	0.9776
6	0.006000	0.9764
7	0.007000	0.9752
8	0.008000	0.9740
9	0.009000	0.9728
10	0.010000	0.9716
11	0.020000	0.9596
12	0.030000	0.9478
13	0.040000	0.9360

Fig. 2.15 Input and Output of TypeCurveGenerator form.

CHAPTER III

METHODOLOGY

The purpose of this study is to develop a computer program that does automatic type curve matching. Four important points have been considered in order to achieve the objectives of this study. The first point is to generate Aminian type curves with minimum input using a computer program. The second point is to synchronize the time of production history data with that of generated curves. The third point is to identify and utilize a comparison method in matching process. The last point is to define a method to iterate reservoir properties.

3.1 CurveGenerator

CurveGenerator is a computer program developed using the same idea of the Sfrac program with fewer input variables. The first part of the Sfrac computer program is used as it is except for some minor changes. The second part of the program is written for an input of F_{NDi} , G_i , and q_i . Sfrac calculates a , b , G_i , F_{NDi} and a type curve for a reservoir where all its properties are known.

Using the conventional method, the coefficients of the deliverability equation (a and b) are calculated using equations (26) and (27). Since, all reservoir properties are not available to calculate a and b using the conventional method, equations (38) and (39) are used. equation (38) and (39) are derived from deliverability equation and non-darcy effect.

$$a = \frac{\Delta P_p}{q_i \cdot F_{NDi}} \quad (38)$$

$$b = \frac{\Delta P_p}{q_i^2} - \frac{a}{q_i} \quad (39)$$

ΔP_p is known because P_{wf} , and P_i are available. Initial flow rate, q_i , is obtained from the production history of the well. To calculate a and b from (38) and (39) respectively, the only variables that has to be assumed is the non-darcy effect (F_{NDi}).

Another important variable is the initial gas-in-place (G_i), which is defined as:

$$G_i = \frac{A \cdot h \cdot \phi \cdot S_g}{B_{gi}} \quad (40)$$

Since the reservoir properties such as reservoir area, formation thickness, porosity, and gas saturation are not available then it is necessary to modify equation (40) in order to use it in generating type curves. The modification is done as follows:

$$A \cdot h \cdot \phi \cdot S_g = Volume \quad (41)$$

$$G_i = \frac{Volume}{B_{gi}} \quad (42)$$

$$Volume = G_i \cdot B_{gi} \quad (43)$$

In equation (43), *Volume* and G_i are the two unknown variables therefore; it is necessary to know one in order to solve for the other one. Overall, it is crucially important to know G_i and F_{NDi} besides the gas properties to generate type curve of a well. CurveGenerator generates Aminian type curve for an input of G_i , q_i , and F_{NDi} . The gas property form and the input & output forms of CurveGenerator are shown in Figure 3.1 and 3.2 respectively.

To verify the results, a curve is generated from Sfrac as part of the output G_i , q_i , and F_{NDi} are obtained. The output of Sfrac is used as an input to CurveGenerator and curve is generated from CurveGenerator as an output. Both type curves, from Sfrac and from CurveGenerator, are plotted in the same graph as shown on Figure 3.3. This verifies that CurveGenerator is generating accurate Aminian type curves.

3.2 Synchronization and Linear Interpolation

Before comparing type curve with production history, the data has to be synchronized. The generated type curves are in different time from that of production history, therefore, linear interpolation is used to find the flow rate of a generated type curve for the list of time in the production history data. Although running time of the program increases by using smaller pressure increment more accurate interpolations can be achieved. Linear interpolation is defined as:

$$q = q_1 + \frac{(q_2 - q_1)}{(t_2 - t_1)}(t - t_1) \quad (44)$$

3.3 Sum of Least Squares Method

Sum of Least Squares method is used to compare type curve with the production history of a well. After the flow rate of type curve is obtained for the production history time then both history flow rate and generated flow rate are compared as follows:

Fig. 3.1 Properties calculation form for CurveGenerator

CurveGenerator

Input and Commands

Pwf (psi)

Gi(Bcf)

Fndi

Qi (Mcf/d)

	Time	Q(Mcf/d)
1	4.53	5478.1
2	9.12	5434.1
3	13.80	5389.9
4	18.53	5345.5
5	23.33	5300.9
6	28.23	5256.1
7	33.21	5211.1
8	38.26	5165.9
9	43.39	5120.5
10	48.61	5074.9
11	53.91	5029.1
12	59.30	4983.1
13	64.79	4936.9
14	70.34	4890.4
15	75.99	4843.8
16	81.75	4796.9
17	87.59	4749.8
18	93.54	4702.5
19	99.59	4655.0

Fig. 3.2 Input and Output of CurveGenerator form.

Fig. 3.3 Comparison of Sfrac and CurveGenerator Decline Curves

$$\sum (q_{original} - q_{generated})^2 \quad (44)$$

For every generated type curve, a sum of least square is stored.

3.4 Iteration of G_i and F_{NDi}

The computer program generates many different type curves using different values of G_i and F_{NDi} in order to match find the best the Production history. Type curves generated for a combination value of G_i and F_{NDi} values are shown in Figure 3.4. The program starts with assumed of F_{NDi} , and an estimated range of G_i . The estimated range is divided into ten evenly distributed G_i values. Using the initial F_{NDi} and the G_i values, ten curves are generated. These ten type curves go through a process of interpolation and comparison. A type curve with the smallest sum of least square is chosen as the best candidate. At this stage, the chosen type curve is the only known type curve that is closer to the production history, then automatically G_i range narrows to new values. The new range is created by moving one step below and above the best G_i value. For example, if the initial range of G_i is from 1 Bcf to 10 Bcf, then the ten candidates will be 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10 Bcf. Suppose the best candidate is the type curve with 5 Bcf, then the new range will be between 4 to 6 Bcf. The process is repeated until the sum of least square is no longer changing. The shape of type curves generated for different values of G_i are shown in Figure 3.5. The value of F_{NDi} is the same for all the curves shown in Figure 3.5. After tolerance is satisfied in G_i iteration, automatically using the best chosen G_i , iteration on F_{NDi} starts with a range of 1.1 to 20. The method used to iterate on F_{NDi} is the same as that of G_i . The process of curve generation, interpolation and comparison is

Fig. 3.4 Plot of Generated Curves for different values of G_i and F_{NDi} .

Fig. 3.5 Plot of Generated Curves for different values of G_i .

Fig. 3.6 Plot of Generated Curves for different values F_{NDi} .

done for ten curves at a time. From the ten curves, the best curve is selected and the program goes iteration process is shown in Figure 3.7. Therefore, the program gives the estimated G_i , F_{NDi} , calculated deliverability coefficients and type curve.

Fig. 3.7 Simple flow chart of G_i and F_{NDi} iteration process.

3.5 Effect of Starting G_i and F_{NDi}

The initial G_i range is the starting point, and its effect to the result is tolerable. It is not necessary for the result to be inside the initial range. The range is to minimize the number of G_i iterations. If the range is broad or if it is far away from the actual value then the number of iteration will be higher.

One of the things that was observed from the results of different runs was that unlike the starting range of G_i , the starting F_{NDi} value had great impact on the final result. Tables 3.1 and Figures 3.8 show how the sum of least squares differs with the starting F_{NDi} . The closer the starting F_{NDi} to the actual F_{NDi} the better result are obtained.

Based on the observation made, it is necessary to use at least ten different starting F_{NDi} in order to cover all options. The best curves obtained for each starting F_{NDi} are considered as the final candidates. Comparing all the final candidates, a reasonable range of F_{NDi} and G_i is achieved. From all the runs made the program was able to choose the best match curve. The computer program that generates Aminian type curves and matches production history to type curves is called MatchCurve. The list of this program is shown on Appendix C. The gas property calculation form and the result form of MatchCurve are shown in Figure 3.9 and 3.10 respectively.

Table 3.1 Run for different Starting F_{NDi} with actual F_{NDi} of 1.54 and actual G_i of 1.9 Bcf

Starting F_{NDi}	G_i	Avg. % Error of Flow rate
1.1	1.45	5.29%
1.25	2	8.93%
1.5	1.9117	0.92%
1.75	1.8615	1.23%
2	1.82	1.49%
2.25	1.7925	1.72%
2.5	1.7657	2.33%
2.75	1.7482	2.08%
3	1.7379	2.16%
3.25	1.7232	2.28%
3.5	1.7107	2.39%

Fig. 3.8 % Error versus Starting F_{NDi} (Actual $G_i = 1.9$ Bcf & $F_{NDi} = 1.54$).

Fig. 3.9 Gas properties calculation Form for MatchCurve Program.

Fig. 3.10 Result Form for MatchCurve Computer Program

CHAPTER IV

RESULTS

MatchCurve program was tested for its accuracy and consistency in matching and predicting gas production. Matches and predictions are verified using three methods. The first method of verification is by plotting the actual curve with the matched type curve. The second method of verification is done by comparing the values of G_i and F_{NDi} of both, the actual and the matched, curves. The third type of verification is by calculating the percent error of the matched flow rate with the corresponding actual flow rate data.

4.1 Matching Results

The test in matching is done for F_{NDi} values ranging from 1.17 to 4.76 and for G_i values ranging from 1.9 to 40 Bcf. These G_i and F_{NDi} values are chosen arbitrarily to cover as wide range as possible to include most reasonable gas wells.

Visual verification of results is analyzed by plotting, the matched type curve with the actual data as shown in Appendix A from Figures A.1 to A.10. In most of the cases, the matched type curve runs on the top of the actual data. In some of the plots, the type curve tends to deviate from the actual points. This deviation mostly occurs near the end of the production history where the production rate is at its lowest stage.

The other verification is calculating the error of matched production rate as shown in Table 4.1 and Figure 4.1. It was observed that the percent error increases with time and ranges from 0.21% to 5.82%. The complete list of Table 4.1 is shown in Appendix as Table A.31. The last seven points of the data have the highest error with

Table 4.1 Match verification

Time(days)	q actual (Mcf/d)	q match (Mcf/d)	% Error
4.53	5478.1	5489.4	0.21%
9.12	5434.1	5446.4	0.23%
13.8	5389.9	5403.2	0.25%
18.53	5345.5	5359.9	0.27%
23.33	5300.9	5316.3	0.29%
28.23	5256.1	5272.5	0.31%
33.21	5211.1	5228.4	0.33%
38.26	5165.9	5184.1	0.35%
43.39	5120.5	5139.7	0.37%
48.61	5074.9	5095.0	0.39%
53.91	5029.1	5050.1	0.42%
59.3	4983.1	5005.0	0.44%
64.79	4936.9	4959.6	0.46%
70.34	4890.4	4914.0	0.48%
75.99	4843.8	4868.2	0.51%

average of 3.9%. This verification also indicates that the biggest percent error appears at the smallest production rate, which is closer to the end of the production history.

The initial gas-in-place and non-darcy factor were also investigated. The average error on initial gas reserve of all the sample runs is 2.07%. The average error on non-darcy factor from all the sample runs is 7.5% as shown in Table 4.2. The deliverability coefficients (a and b) are calculated using the estimated non-darcy factor. Deliverability coefficient, b, has an error of 24.5% in some cases. The percent error seems very high but the effect to the outcome is minimal.

In general, the overall result in matching is reasonable. More than 90% of the matched data has an average error less than 1%. The small deviation that appears at the end of the data is negligible because the amount of gas recovered at that rate is less than 0.05% of the total gas in-place.

Table 4.2 G_i and F_{NDi} of actual and matched Curves

	Actual	Matched	% Error
G_i (Bcf)	4.5	4.498	0.05%
F_{NDi}	2	2.186	8.52%

Figure 4.1 Actual ($F_{NDi} = 2$) and Matched Curves

4.2 Results On Predicting Gas Production

Gas well prediction is the ultimate aim of type curve analysis. Gas wells production is forecasted based on the match. Two prediction samples are briefed for analysis.

4.2.1 Sample One

This sample has a production history up to 1243 days (3.4 years). The initial gas-in-place and non-darcy factor of the well are 5.84 Bcf and 1.59 respectively. The production history is cut to 504 days (1.4 years) and provided as an input to MatchCurve program in order to predict the rest of the actual data. . Type curve is selected based on the match made for the input production history (1.4 years). The actual data (1.4 years) and the type curve (3.4 years) are plotted as shown in Figure 4.2. The type curve goes beyond the production history and predict future production of the well. The plot of the total actual data (3.4 years) and the type curve (3.4 years) are also shown in Figure 4.3.

The predicted and actual production rates are plotted in a Cartesian scale for better analysis as shown in Figure 4.4. The plot shows that predicted production rate runs very close to the actual data. The error of the predicted production rates ranges from 1.58% to 5.65% for two years prediction as shown in Table 4.3 The average error for the first year's prediction was calculated as 3.27% and 5.4% for the second year's of prediction.

The other parameters that were analyzed are the initial gas in-place and the non-darcy factor of the actual and predicting curve as shown in Table 4.4. The initial gas in-place and non-darcy factor of the predicting curve are equal to 5.8334 Bcf and 1.606 and the errors are calculated as 0.11% and 1% respectively. The deliverability coefficients (a

and b) are calculated using the estimated non-darcy factor. The error on deliverability coefficient, b, is 24.5%. The reason for this big error is the way MatchCurve and Sfrac calculates the value of b is slightly different. The effect of this error to the overall result is tolerable. Overall, the result of this sample is very satisfactory.

Table 4.3 Gas prediction of sample one

Time (days)	Actual q (Mcf/d)	Predicted q (Mcf/d)	% Error
504.0	2700.4	2657.7	1.58%
521.0	2567.0	2520.5	1.81%
538.8	2433.7	2383.8	2.05%
557.6	2300.6	2247.6	2.30%
577.6	2167.5	2112.2	2.55%
598.7	2034.6	1977.2	2.82%
621.3	1901.9	1843.1	3.09%
645.5	1769.3	1709.7	3.37%
671.6	1636.8	1577.0	3.65%
699.9	1504.6	1445.3	3.94%
730.9	1372.5	1314.4	4.23%
764.9	1240.6	1184.5	4.52%
803.0	1108.9	1055.6	4.81%
845.9	977.5	927.8	5.08%
895.4	846.3	801.2	5.32%
953.7	715.3	675.7	5.54%
1024.8	584.6	551.6	5.65%
1115.8	454.1	428.9	5.55%
1242.8	324.0	308.0	4.93%

Table 4.4 G_i and F_{NDi} results of sample one

	Actual	Match	% Error
G_i (Bcf)	5.8399	5.8334	0.11%
F_{NDi}	1.59	1.606	1.01%
A	27049	26750.1	1.11%
B	1.63	1.23	24.54%

Figure 4.2 Actual data (1.4 years) and predicting curve (3.4 years)

Figure 4.3 Total Actual data and predicting curve of sample one

Figure 4.4 Prediction of gas production for 2 years.

4.2.2 Sample two

The actual production of this sample is 2 years longer than the first sample. The production history available for this well is up to 1965 days (5.4 years). The initial gas in-place and non-darcy factor of the well are 22.92 Bcf and 1.88 respectively. The production history is cut to 821 days (2.25 years) and used as input to MatchCurve to predict the production for the remaining 1144 days (3.15 years). The plot of actual data (2.25 years) and the predicting type curve (5.4 years) are shown in Figure 4.5. The actual data (5.4 years) and the type curve (5.4 years) are plotted is shown in Figure 4.6. It is hard to detect the deviation in a log-log plot therefore the prediction data obtained as result from the match and the actual data are plotted in a Cartesian plot as shown Figure 4.7. The predicted data is higher than the actual data towards the end of the curve. The type curve starts to deviate from the actual data with time. The error of the predicted production rate is calculated as shown in Table 4.5. The average error for the first, second and third years of gas prediction were calculated as 3.56%, 10.23% and 21.82% respectively. The more data is cut from the actual data the less accurate the prediction will be because there will be more than one type curves that will fit the input data. Type curves, with the same initial gas in-place, bend down wards at different time depending on the value of F_{NDi} . The bigger the F_{NDi} value the longer it takes to bend and the bigger data it takes to get accurate match.

The initial gas-in-place and non-darcy factor of the predicting curve are equal to 23.63 Bcf and 1.62 with error of 3.1% and 13.6% respectively. The error of the non-darcy factor affects the deliverability coefficients (a and b). The calculated deliverability coefficients (a and b) have an error of 13% and 17% respectively as shown in Table 4.6. Deliverability coefficient, a, has a bigger impact on the overall result. The error causes

the deviation of the predicting curve from the actual data near the end of the production history.

Table 4.5 Gas prediction of sample two

Time(days)	q actual (Mcf/d)	q predicted (Mcf/d)	% Error
821.93	6260.9	6381.2	1.92%
850.63	5894.3	6018.9	2.11%
881.14	5526.5	5656.7	2.36%
913.75	5157.4	5293.9	2.65%
948.79	4787.0	4930.5	3.00%
986.63	4415.2	4566.7	3.43%
1027.84	4041.9	4201.6	3.95%
1073.12	3666.9	3835.1	4.59%
1123.41	3290.3	3466.7	5.36%
1180.01	2911.8	3096.1	6.33%
1244.90	2531.3	2721.7	7.52%
1321.18	2148.8	2341.9	8.98%
1413.56	1764.0	1956.2	10.89%
1531.41	1376.8	1563.1	13.53%
1695.10	987.0	1163.4	17.88%
1965.87	594.4	747.5	25.76%

Table 4.6 G_i and F_{NDi} results of sample one

	Actual	Predicted	% Error
G_i (Bcf)	22.92	23.63	3.1%
F_{NDi}	1.88	1.62	13.6%

Figure 4.5 Actual data (2.25 years) and predicting curve (5.4 years)

Figure 4.6 Total actual data and predicting curve of sample two

Figure 4.7 Prediction of gas production for 3.2 years

CHAPTER V

DISCUSSION

This chapter discusses the effect of gas reserve and non-darcy factor in matching production rates and in predicting gas wells. The type curves sensitivity to non-darcy factor and MatchCurve program are also briefly discussed.

5.1 G_i and F_{NDi} in Curve Matching

The initial gas in-place and non-darcy factor of different matches are analyzed as shown in Table 5.1. The percent error on the initial gas in-place of all the runs was below 10%. For non-darcy factor less than two, the estimated initial gas in-place error ranges from 0.1% to 3.28%. The range of error for non-darcy factor greater than two was from 1.21% to 9.64%. The variation of the actual initial gas in-place has no effect on the fluctuation of the percent error. Overall, the result in estimating initial gas in-place is satisfactory.

The variation of actual non-darcy factor has greater effect on the estimated non-darcy factor. For actual non-darcy factor less than two, the error of the estimated non-darcy factor ranges from 0.13% to 10.39%. For actual non-darcy factor greater than two, the error ranges from 11.86% to 27.18%. This error is directly proportional to actual non-darcy factor. The correlation between permeability and non-darcy factor of a gas well is shown in Table 5.2. Non-darcy factor is directly proportional to formation permeability and most common gas wells have tight formations and their permeability is usually less than 20 md. Since matching results for actual non-darcy factor less than two are reasonable this program can be used for most of the real production history cases.

Table 5.1 Sample runs in matching

Run No.	Actual G_i (Bcf)	Estimated G_i (Bcf)	Actual F_{NDi}	Estimated F_{NDi}	% Error of G_i	% Error of F_{NDi}
1	1.900	1.934	1.17	1.21	1.79%	3.58%
2	13.8	13.786	1.25	1.34	0.10%	7.00%
3	2.665	2.752	1.46	1.39	3.28%	5.07%
4	40.490	40.597	1.55	1.57	0.26%	1.35%
5	24.917	24.999	1.57	1.58	0.33%	0.45%
6	18.168	18.244	1.58	1.58	0.42%	0.13%
7	5.293	5.405	1.81	1.62	2.10%	10.39%
8	1.777	1.605	2.2	2.46	9.64%	11.86%
9	2.800	2.869	3.02	2.45	2.45%	18.87%
10	1.290	1.306	4.76	3.47	1.21%	27.18%

Table 5.2 Correlation of Permeability and F_{NDi}

K (md)	F_{NDi}
1	1.12
20	2.00
50	2.68
100	3.44

5.2 Prediction of Gas Wells

The average error of production rate in predicting for the first year and second year of gas production are equal to 3.4% and 8% respectively. The error increases as predicting time increases depending on the gas reservoir. One of the reasons that the error near the end of production increase is that the production rate closer to the end is small therefore a small error in prediction is calculated as large percent. It is significant to have a lesser error at higher production rate than near end of the production. The limit on how much gas production can be predicted and what percent should be produced before prediction are shown in Table 5.3. The percent error on initial gas in-place, non-darcy factor and predicted rate increases as the amount of gas production to be predicted

increases. As the table shows, 11% of the total reserve was forecasted with accurate estimation of initial gas in-place and non-darcy factor. 30% of the total reserve was forecasted but the percent error on the initial gas in-place and non-darcy factor was high. The plot of the actual data and the predicting curve is shown in Figure 5.1. The type curve starts to separate from the actual data after 300 days of production while the production history used as input was up to 204 days. Therefore, the type curve predicts accurately for 96 days only. One of the main reasons for this limitation is that the effect of non-darcy factor on the type curves does not appear until that the type curve starts to bend down wards. Therefore trying to predict, before the curve starts to bend, will cause bigger error in estimating non-darcy factor and in forecasting future gas production.

Table 5.3 Sample runs in predicting gas production

Run No.	Prediction time (years)	% recovered History Data	% recovered After Prediction	Net % Predicted	% Error G_i	% Error F_{NDi}
11	0.00	68.22%	68.22%	0	0.66%	0.63%
12	0.9	66.82%	68.23%	1.41%	0.66%	0.63%
13	1.75	63.34%	68.23%	4.89%	0.56%	0.31%
14	2.5	57.19%	68.23%	11.04%	0.11%	1.01%
15	2.01	39.73%	70.35%	30.62%	14.52%	213.96%

5.3 Sensitivity of Type Curve

Non-darcy factor changes the shape of type curves while initial gas in-place represents the area under the type curve. Type curves are sensitive to non-darcy factor. Figure 3.6 in chapter three, shows type curves with difference of 1 non-darcy factor. It is hard to imagine how the curves will look like if the difference between the curves was equal to $0.05 F_{NDi}$. It would be almost impossible to differentiate them with naked eyes.

For actual non-darcy factor of two, if the difference between the matched and actual non-darcy factor is 0.05 then the error is 2.5%. It is almost impossible to estimate non-darcy factor accurately by superimposing a production history on the top of the type curves manually. The deliverability coefficients (a and b) are calculated using the estimated non-darcy factor. Deliverability coefficient, a, has more impact on production rate relative to b as shown in equations (37) and (38).

5.4 Brief Discussion about MatchCurve

Thousands of type curves can be generated with different combination of non-darcy factor and initial gas in-place. MatchCurve computer program steers itself to the correct direction until it determines the best match. In one run, the computer program compares an average of 2000 to 4000 type curves before the best match type curve is selected. The program also interpolates for every data of each curve, which is approximately 200,000 to 400,000, data per run.

Figure 5.1 Predicting production with small data.

CHAPTER VI

CONCLUSIONS

The following conclusions are made from the study:

- A computer program is developed to generate accurate Aminian type curves with non-darcy factor, initial gas reserve and gas properties.
- The developed computer program synchronizes the values of time for the generated type curve with the production history for comparison process.
- The computer program compares the production history with a type curve and best match is achieved.
- Matches are verified by plotting the production history and the best match curve on the same graph.
- The average error encountered with the best-matched curve were generally less than 2%. The error is calculated by comparing the production rates from the type curve and the production history.
- Initial gas in-place, and non-darcy factor for a number of simulated the production history were estimated with an average error of 2.07% and 7.5% respectively.
- More accurate results were achieved in estimating non-darcy factor for production history generated with a non-darcy factor smaller than 2. The average error for these cases was less than 4%.
- The coefficient of the deliverability equation (a and b) are calculated from estimated non-darcy factor.
- Gas production was predicted for a year beyond the history with an average error of 3.4%. The average error for the second year prediction was equal 8%. The

error was calculated by comparing the production rate of the actual data and the predicted values.

CHAPTER VII

RECOMMENDATIONS

- In conventional calculation, the value of deliverability coefficient, b changes as the average viscosity changes during production. For example in Sfrac b changes from 2.221 to 2.23 in generating one type curve. In MatchCurve, b is constant per run (per type curve), that could cause small error in the final result. Therefore by running Sfrac for different samples and watching how b varies it is possible to find a correlation that can be implemented to MatchCurve.
- MatchCurve uses Sfrac gas simulator to generate production history and all the tests has been done using the simulator. Testing MatchCurve for real production history would be one the recommendations for future.
- The program can also be modified for smaller ranges of non-darcy factor to achieve better accuracy in determine the non-darcy factor of the best match.

REFERENCES

1. Koederitz, L.F., Harvey, A. H., Honarpour, M., Introduction to Petroleum Reservoir Analysis," Published by Gulf Publishing Company, Houston, TX. 1989.
2. Bradley, B.H., Petroleum Engineering handbook, Third Printing, SPE, Richardson, TX. 1992.
3. Slider, H.C., Practical Petroleum Reservoir Engineering Methods -An Energy Conservation Science," Petroleum Publishing Company, Tulsa, Oklahoma 1976.
4. Chierici, L.G., Principle of Petroleum Reservoir Engineering, trans. Peter J.W. New York: 1995
5. Arps, J.J., "Analysis of Decline Curves," Trans., AIME, 1945. P. 160, 228 -247.
6. Fetkovich, M.J., "Decline Curve Analysis Using Type Curves," Presented SPE-AIME, Las Vegas, Nevada. 1973. (SPE Serial No. 4629)
7. Gringarten, C.A., "Type-Curve Analysis: What It Can Do and Can Not Do," JPT, January 1987.
8. Fetkovich, M.J., Vienot, M.E., Bradley, M.D., Kiesow, U.G., "Decline Curve Analysis Using Type Curve- Case Histories," SPE Formation Evaluation, December 1987, P.637-656.
9. Lee, J., Wattenbarger, A.R., Gas Reservoir Engineering, Henry L. Doherty Memorial Fund of AIME-SPE, SPE Volume 5., Richardson, TX. 1996.
10. Carter, R.D., "Type Curve for Finite Radial and Linear Gas Flow System: Constant-Terminal-Pressure Case," SPEJ, Oct. 1985.

11. Fraim, M.L., Wattenbarger, R.A., " Gas Reservoir Decline-Curve Analysis Using Type Curves With Real Gas Pseudo-pressure And Normalized Time," paper SPE Serial No. 14238.
12. Aminian, K., Ameri, S., Hyman, M., " Production Decline Type Curve For Gas Wells Producing Under Pseudo Steady State Condition," paper SPE 15933 presented at 1986 SPE Eastern Regional Meeting, Columbus, OH, Nov. 12-14
13. Schmid, S.H., Caudle, B.H., and Miller, M. A., "Gas Well Decline Analysis Incorporating Real Gas Behavior and Non-Darcy Flow," paper SPE 15521 presented at the 1986 SPE Annual Technical Conference and Exhibition, New Orleans, Oct. 5-8.
14. Aminian, K., Ameri, S., Beg, N., Yost II, A.B., "Production Forecasting of Gas Wells Under Altered Conditions," Presented at SPE Eastern Regional Meeting, Pittsburgh, PA. September 1987
15. Aminian, K., Ameri, S., Stark, J.J., Yost II, A.B., "Gas Well Production Decline In Multiwell Reservoirs," JPT, December 1990, P.1573-1579.
16. Dascalescu, B.W., "Production Behavior of Gas Wells During The Unsteady State Period A Type Curve Approach," Master's Thesis, West Virginia U., Morgantown, WV. 1994.
17. Abidi, R.H., "The Effect of Pressure Dependent Gas Properties on the Performance of Gas Well," Morgantown, WV. Masters Thesis, 1992.
18. Aminian, K., Ameri, S., Puon, P.S., and Rieke, H.H., "Production History Analysis in Predicting 'Dry' Gas-Well Performance," Journal of Science and Engineering, 1989. 13-26.

APPENDIX A

SAMPLE RUNS

Table A.1 Actual data ($F_{NDi}=1.17$ and $G_i = 1.9$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
4.64	2271.63	4.69	2273.632
9.34	2246.9	9.45	2250.35
14.16	2222.16	14.32	2227.019
19.05	2197.41	19.26	2203.638
24.01	2172.64	24.27	2180.208

Table A.2 MatchCurve results for actual data ($F_{NDi} = 1.17$ and $G_i = 1.9$ Bcf)

	Actual	Min	Max	Best
$G_i(\text{Bcf})$	1.9	7.8278	8.1524	1.934
F_{NDi}	1.17	1.385	1.579	1.21

Fig. A.1 Visual verification of match for actual data ($F_{NDi} = 1.17$ and $G_i = 1.9$)

Table A.3 Actual data ($F_{NDi}=1.25$ and $G_i = 13.8$ Bcf) and best match

Time(days)	Actual q(Mcfd)	Time (days)	Matched q(Mcfd)
10.22	7454.99	10.19	7463.906
20.59	7376.56	20.52	7392.172
31.19	7298.05	31.07	7320.223
41.95	7219.46	41.78	7248.06
52.88	7140.8	52.63	7175.684

Table A.4 MatchCurve results for actual data ($F_{NDi}=1.25$ and $G_i = 13.8$ Bcf)

	Actual	Min	Max	Best
G_i	13.8	13.78631	18.9988	13.78631
F_{NDi}	1.25	1.385	1.578	1.337556
a	60493	13737.1	15656.5	56326.53
b	2.6	0.23	0.3	2.535207

Fig. A.2 Visual verification of match for actual data ($F_{NDi}=1.25$ and $G_i = 13.8$ Bcf)

Table A.5 Actual data ($F_{NDi}=1.46$ and $G_i = 2.665$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
1.73	9141.33	1.78	9150.797
3.48	9048.08	3.59	9059.934
5.26	8954.78	5.43	8968.867
7.07	8861.39	7.29	8877.599

Table A.6 MatchCurve results for actual data ($F_{NDi}=1.46$ and $G_i = 2.665$ Bcf)

	Actual	Min	Max	Best
G_i	2.665	2.6375	2.7523	2.7523
F_{NDi}	1.46	1.386	1.605	1.386
a	40272	36652.9	42433.7	42433.7
b	2.52	1.78	2.41	1.78

Fig. A.3 Visual verification of match for actual data ($F_{NDi}=1.46$ and $G_i = 2.665$ Bcf)

Table A.7 Actual data ($F_{NDi}=1.55$ and $G_i = 40.49$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
7.3	30580.48	7.31	30628.955
14.71	30285.18	14.72	30358.421
22.27	29989.4	22.28	30086.848
29.95	29693.13	29.94	29814.235
37.73	29396.38	37.71	29540.578
45.7	29099.04	45.65	29265.874

Table A.8 MatchCurve results for actual data ($F_{NDi}=1.55$ and $G_i = 40.49$ Bcf)

	Actual	Min	Max	Best
G_i	40.49	39.4813	40.5971	40.5971
F_{NDi}	1.55	1.571	1.742	1.571
a	11904	10544.9	11693.5	11693.5
b	0.28	0.22	0.25	0.22

Fig. A.4 Visual verification of match for actual data ($F_{NDi}=1.55$ and $G_i = 40.49$ Bcf)

Table A.9 Actual data ($F_{NDi}=1.57$ and $G_i = 24.9$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
4.44	30981.28	4.44	31030.658
8.93	30683.52	8.95	30757.113
13.53	30385.27	13.54	30482.513
18.19	30086.52	18.2	30206.856
22.92	29787.28	22.92	29930.139
27.76	29487.43	27.75	29652.359

Table A.10 MatchCurve results for actual data ($F_{NDi}=1.57$ and $G_i = 24.9$ Bcf)

	Actual	Min	Max	Best
G_i	24.9169	24.9992	25.8735	24.9992
F_{NDi}	1.57	1.402	1.577	1.577
a	11582	11493.6	12929.3	11493.6
b	0.28	0.17	0.21	0.21

Fig. A.5 Visual verification of match for actual data ($F_{NDi}=1.57$ and $G_i = 24.9$ Bcf)

Table A.11 Actual data ($F_{NDi}=1.58$ and $G_i = 18.168$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
3.87	25910.73	3.88	25952.048
7.79	25662.03	7.81	25723.316
11.8	25412.92	11.82	25493.702
15.86	25163.39	15.88	25263.204
19.98	24913.45	20	25031.819
24.2	24663	24.21	24799.544

Table A.12 MatchCurve results for actual data ($F_{NDi}=1.58$ and $G_i = 18.168$ Bcf)

	Actual	Min	Max	Best
G_i	18.168	18.2444	18.9988	18.2444
F_{NDi}	1.58	1.385	1.578	1.578
a	13792.5	13737.1	15656.5	13737.1
b	0.41	0.23	0.3	0.3

Fig. A.6 Visual verification of match for actual data ($F_{NDi}=1.58$ and $G_i = 18.168$ Bcf)

Table A.13 Actual data ($F_{NDi}=1.81$ and $G_i = 5.29$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
1.62	14165.78	1.64	14201.395
3.26	14060.72	3.32	14102.47
4.92	13955.5	5.01	14003.213
6.61	13850.11	6.73	13903.623
8.31	13744.54	8.46	13803.699

Table A.14 MatchCurve results for actual data ($F_{NDi}=1.81$ and $G_i = 5.29$ Bcf)

	Actual	Min	Max	Best
G_i	5.2934	5.2904	5.4045	5.4045
F_{NDi}	1.81	1.622	1.783	1.622
a	27687	27916.2	30683.7	30683.7
b	2.51	1.34	1.53	1.34

Fig. A.7 Visual verification of match for actual data ($F_{NDi}=1.81$ and $G_i = 5.29$ Bcf)

Table A.15 Actual data ($F_{NDi}=2.2$ and $G_i = 1.77$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
0.45	23145.47	0.31	23200.459
0.92	22894.48	0.63	23032.325
1.39	22642.86	0.96	22863.454
1.86	22390.5	1.29	22693.839
2.35	22137.38	1.62	22523.474
2.84	21883.49	1.95	22352.351

Table A.16 MatchCurve results for actual data ($F_{NDi}=2.2$ and $G_i = 1.77$ Bcf)

	Actual	Min	Max	Best
G_i	1.7767	1.5829	1.6054	1.6054
F_{NDi}	2.2	2.461	2.673	2.461
a	8710	9924.9	10780.2	10780.2
b	0.52	0.68	0.71	0.68

Fig. A.8 Visual verification of match for actual data ($F_{NDi}=2.2$ and $G_i = 1.77$ Bcf)

Table A.17 Actual data ($F_{NDi}=3.02$ and $G_i = 12.8$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Matched q (Mcf/d)
0.39	43296.51	0.4	43377.583
0.79	42923.57	0.81	43019.596
1.2	42550.28	1.22	42660.453
1.61	42176.4	1.64	42300.151
2.02	41801.91	2.06	41938.684
2.44	41427.06	2.49	41576.048

Table A.18 MatchCurve results for actual data ($F_{NDi}=3.02$ and $G_i = 12.8$ Bcf)

	Actual	Min	Max	Best
G_i	2.8	2.8267	2.8687	2.8687
F_{NDi}	3.02	2.45	2.705	2.45
a	3932	4362.7	4816.8	4816.8
b	0.21	0.16	0.17	0.16

Fig. A.9 Visual verification of match for actual data ($F_{NDi}=3.02$ and $G_i = 12.8$ Bcf)

Table A.19 Actual data ($F_{NDi}=4.76$ and $G_i = 1.29$ Bcf) and best match

Time (days)	Actual q (Mcf/d)	Time (days)	Match q (Mcf/d)
0.18	79935.85	0.18	79839.443
0.35	78718.07	0.36	78683.503
0.54	77496.4	0.54	77519.697
0.72	76271.25	0.73	76347.876
0.91	75041.91	0.92	75167.883

Table A.20 MatchCurve results for actual data ($F_{NDi}=4.76$ and $G_i = 1.29$ Bcf)

	Actual	Min	Max	Best
G_i (Bcf)	1.29	1.3056	1.3172	1.3056
F_{NDi}	4.76	3.259	3.466	3.466
a	636	871	926.2	871
b	0.03	0.03	0.03	0.03

Fig. A.10 Visual verification of match for actual data ($F_{NDi}=4.76$ and $G_i = 1.29$ Bcf)

Table A.21 Actual data (1453 days) and predicting curve (1444 days)

Time (days)	Actual q (Mcf/d)	Time (days)	Match q (Mcf/d)
2.46	13073.63	2.48	13094.517
4.96	12948.57	4.99	12979.186
7.51	12823.3	7.55	12863.409
10.1	12697.81	10.14	12747.186

Table A.22 Results for matched type curve

	Actual	Min	Max	Best
G_I	5.8399	5.8783	6.1214	5.8783
F_{NDi}	1.59	1.386	1.58	1.58
a	27049	27185.3	30999.4	27185.3
b	1.63	0.91	1.2	1.2

Fig. A.11 Plot of production history and type curve

Table A.23 Actual data (1115 days) and predicting curve (1444 days)

Time (days)	Actual q (Mcfd)	Time (days)	Match q (Mcfd)
2.46	13073.63	2.48	13094.517
4.96	12948.57	4.99	12979.187
7.51	12823.3	7.55	12863.41
10.1	12697.81	10.14	12747.187
12.73	12572.12	12.77	12630.517
15.41	12446.16	15.46	12513.397

Table A.24 Results for predicting for 329 days

	Actual	Min	Max	Best
$G_i(\text{Bcf})$	5.8399	5.8783	6.1234	5.8783
F_{NDi}	1.59	1.385	1.58	1.58
A	27049	27185.1	31028.9	27185.1
B	1.63	0.91	1.2	1.2

Fig. A.12 Plot of predicting curve (329 days of prediction) and history data

Table A.25 Actual data (802 days) and predicting curve (1440 days)

Time (days)	Actual q (Mcfd)	Time (days)	Match q (Mcfd)
2.46	13073.63	2.47	13094.601
4.96	12948.57	4.98	12979.437
7.51	12823.3	7.54	12863.826
10.1	12697.81	10.13	12747.768

Table A.26 Results for predicting for 638 days

	Actual	Min	Max	Best
$G_i(\text{Bcf})$	5.8399	5.8724	6.1313	5.8724
F_{NDi}	1.59	1.382	1.585	1.585
a	27049	27099.4	31078.4	27099.4
b	1.63	0.9	1.21	1.21

Fig. A.13 Plot of predicting curve (638 days of prediction) and history data

Table A.27 Actual data (504 days) and predicting curve (1417 days)

Time (days)	Actual q (Mcf/d)	Time (days)	Match q (Mcf/d)
2.46	13073.63	2.46	13094.939
4.96	12948.57	4.95	12980.449
7.51	12823.3	7.49	12865.51
10.1	12697.81	10.06	12750.12

Table A.28 Results for predicting for 913 days

	Actual	Min	Max	Best
G_I (Bcf)	5.8724	5.6045	5.8334	5.8334
F_{NDi}	1.382	1.606	1.826	1.606
A	27099.4	23532.8	26750.1	26750.1
B	0.9	1.23	1.48	1.23

Fig. A.14 Plot of predicting curve (736 days of prediction) and history data

Table A.29 Actual data (207 days) and predicting curve (943 days)

Time (days)	Actual q (Mcf/d)	Time (days)	Match q (Mcf/d)
2.46	13073.63	2.1	13102.518
4.96	12948.57	4.23	13003.175
7.51	12823.3	6.4	12903.357
10.1	12697.81	8.59	12803.062

Table A.30 Results for predicting for 736 days

	Actual	Min	Max	Best
G_I	5.8724	4.8725	4.992	4.992
F_{NDi}	1.382	2.431	2.714	2.431
a	27099.4	15830.9	17671.5	17671.5
b	0.9	1.92	2.06	1.92

Fig. A.15 Plot of predicting curve (736 days of prediction) and history data

Table A.31 Match verification

Time(days)	q actual (Mcf/d)	q match (Mcf/d)	% Error
4.53	5478.1	5489.4	0.21%
9.12	5434.1	5446.4	0.23%
13.8	5389.9	5403.2	0.25%
18.53	5345.5	5359.9	0.27%
23.33	5300.9	5316.3	0.29%
28.23	5256.1	5272.5	0.31%
33.21	5211.1	5228.4	0.33%
38.26	5165.9	5184.1	0.35%
43.39	5120.5	5139.7	0.37%
48.61	5074.9	5095.0	0.39%
53.91	5029.1	5050.1	0.42%
59.3	4983.1	5005.0	0.44%
64.79	4936.9	4959.6	0.46%
70.34	4890.4	4914.0	0.48%
75.99	4843.8	4868.2	0.51%
81.75	4796.9	4822.2	0.53%
87.59	4749.8	4775.9	0.55%
93.54	4702.5	4729.4	0.57%
99.59	4655.0	4682.7	0.59%
105.72	4607.3	4635.8	0.62%
111.98	4559.3	4588.5	0.64%
118.34	4511.1	4541.1	0.66%
124.81	4462.7	4493.4	0.69%
131.38	4414.1	4445.5	0.71%
138.09	4365.3	4397.3	0.73%
144.91	4316.2	4348.9	0.76%
151.83	4267.0	4300.3	0.78%
158.89	4217.4	4251.4	0.81%
166.07	4167.7	4202.3	0.83%
173.38	4117.7	4152.9	0.85%
180.83	4067.6	4103.2	0.88%
188.4	4017.1	4053.3	0.90%
196.12	3966.5	4003.2	0.93%
203.98	3915.6	3952.8	0.95%
211.98	3864.5	3902.1	0.97%
220.14	3813.2	3851.2	1.00%
228.45	3761.6	3800.0	1.02%
236.91	3709.8	3748.6	1.05%
245.54	3657.8	3696.9	1.07%
254.33	3605.5	3644.9	1.09%
263.29	3553.0	3592.7	1.12%
281.74	3447.3	3487.5	1.16%
291.24	3394.2	3434.4	1.19%

Time (days)	q Actual (Mcf/d)	q match (Mcf/d)	% Error
300.93	3340.7	3381.2	1.21%
310.82	3287.1	3327.6	1.23%
320.9	3233.2	3273.8	1.26%
331.21	3179.1	3219.6	1.28%
341.72	3124.7	3165.2	1.30%
352.43	3070.1	3110.6	1.32%
363.4	3015.3	3055.7	1.34%
374.6	2960.2	3000.5	1.36%
386.05	2904.9	2944.9	1.38%
397.74	2849.4	2889.2	1.40%
409.7	2793.6	2833.1	1.41%
421.93	2737.6	2776.7	1.43%
434.44	2681.4	2720.1	1.44%
447.25	2625.0	2663.2	1.46%
460.36	2568.3	2606.0	1.47%
473.8	2511.3	2548.5	1.48%
487.57	2454.2	2490.7	1.49%
501.67	2396.8	2432.7	1.50%
516.15	2339.2	2374.3	1.50%
531	2281.3	2315.7	1.51%
546.26	2223.2	2256.7	1.51%
561.94	2164.9	2197.4	1.50%
578.05	2106.3	2137.9	1.50%
594.64	2047.6	2078.0	1.49%
611.7	1988.5	2017.9	1.47%
629.3	1929.3	1957.4	1.46%
647.46	1869.8	1896.6	1.43%
666.19	1810.1	1835.6	1.41%
685.56	1750.1	1774.2	1.37%
705.61	1690.0	1712.5	1.33%
726.36	1629.5	1650.5	1.29%
747.9	1568.9	1588.2	1.23%
770.28	1508.0	1525.6	1.17%
793.57	1446.8	1462.7	1.09%
817.85	1385.5	1399.4	1.01%
843.22	1323.9	1335.9	0.91%
869.78	1262.0	1272.0	0.79%
897.65	1199.9	1207.8	0.66%
926.98	1137.5	1143.3	0.51%
957.94	1074.9	1078.6	0.34%
990.75	1012.0	1013.5	0.14%
1025.63	948.9	948.2	0.08%
1062.92	885.5	882.5	0.33%
1102.98	821.8	816.6	0.63%
1146.26	757.9	750.5	0.97%

Time (days)	q Actual (Mcf/d)	q match (Mcf/d)	% Error
1193.4	693.7	684.2	1.36%
1245.2	629.2	617.8	1.81%
1302.76	564.3	551.2	2.33%
1367.57	499.2	484.7	2.92%
1441.89	433.8	418.3	3.57%
1529.32	368.0	352.0	4.35%
1635.26	302.0	286.4	5.15%
1770.51	235.5	221.8	5.82%

APPENDIX B

LIST OF SFRAC

'all variables must be declared
Option Explicit

Global GasComResult, tit1, tit2, tit3, tit4, tit5, tit6, tit7, Chart1, chart2, Say As String
Global K(10), S(10), H(10), CA(10), Area(10), Pwf(10), Rw(10), Sg(10), porosity(10) As Single
Global Zwf(10), Xi(10), M(20), Pa(20), T(20) As Single
Global Pre, NSteps, MaxP, Temp, G, DeltaP, Pi, Pnow As Single
Global PRead(4001), ZRead(4001), MPRead(4001), VRead(4001) As Single
Global R, theindex, count3, count2 As Integer
Global Znow, MP, ViscR, ViscWF, ViscAvg, MPWF, DeltaMP, Re As Single
Global Area1, Term, a, b, Beta, B1, MQ, Q, Qnow, Qi, Qavg As Single
Global Zi, Bgi, Bgf, Gi, GpScf, GpMcf, CumGpScf, TDays As Single
Global Tyears, CumYears, Dq, DGp, DT, CumDays, Z9 As Single
Global DTD, XSpan, iGp, QiQ, X1(10, 4001), X2(10, 4001), X3(10, 4001), X4(10, 4001), X5(10, 4001) As Single
Global X6(10, 4001), X7(10, 4001), X5a(10, 4001), RPressure(10, 4001), RDT(10, 4001) as Single
Global RDQ(10, 4001), RDGP(10, 4001), RQavg(10, 4001), RCumDays(10, 4001), RCumGPscf(10, 4001)
Global i, j, N1, N3, gflag, R1 As Integer
Global Dif, Diff, Diff1, ratio As Single
Global P1, P2, Z1, Z2, MP1, MP2, Visc1, Visc2 As Single
Global delX, delY1, delY2, DelX1, DelDY1, DelDY2 As Long
Global MaxProd, maxQavg, maxTime, MaxDt, MaxDGp, MaxDq, Fndi As Single
Global Gasg, Tpc, Ppc As Single
Global PathName\$, FLNM\$, pathname1\$, FLNM1\$, GasCompositionResult As String
Dim Ma, G1, G2, G3, G4, Cpr, count, P As Single
Dim Zfactor, Compressibility, Viscosity As Single
Dim Tmp, tmp1, tmp2, tmp3, tmp4, DeltaPressure As Single
Dim PseudoPressure As Long
Dim press(4001), Zfa(4001), Comp(4001), Visc(4001), PseudoP(4001) As Single

```
Private Sub cmdExit_Click()
End
End Sub
```

```
Private Sub cmdGasComposition_Click()
frmGasComposition.Show
End Sub
```

```
Private Sub cmdGoToDeclineCurve_Click()
frmDeclineCurve.Show
End Sub
```

```
Private Sub cmdPrint_Click()
frmGasProperties.PrintForm
End Sub
```

```
Private Sub cmdRun_Click()
Dim A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11 As Single
Dim j1, j2, j3, j4, count As Integer
Dim D, E, b, C, Density, deltax, deltax As Single
Dim Gasg, Zfactor, Compressibility, Viscosity, Ppc, Ppr, Tpr, Tpc, D5, Z As Single
Dim Ma, G1, G2, G3, G4, Cpr, P As Single
Dim Tmp, tmp1, tmp2, tmp3, tmp4, DeltaPressure As Single
Dim PathName$, FLNM$, GasCompositionResult As String
Dim PseudoPressure As Long
Dim press(4001), deltaPress(4001), Zfa(4001), Comp(4001), Visc(4001), PseudoP(4001) As Single
Picture1.Cls
gflag = 0
PseudoPressure = 0
Tmp = 0
Let A1 = 0.3265: A2 = -1.07: A3 = -0.5339
A4 = 0.01569: A5 = -0.05165: A6 = 0.5475
```

```

A7 = -0.7361: A8 = 0.1844: A9 = 0.1056
A10 = 0.6134: A11 = 0.721
count = 1
If gflag = 0 Then
    Ppc = 709.604 - 58.718 * Val(TxtGasg)
    Tpc = 170.491 + 307.344 * Val(TxtGasg)
Else
    Ppc = Val(TxtPpc)
    Tpc = Val(TxtTpc)
    TxtGasg.Text = Gasg
End If
Tpr = (Val(TxtTemp) + 460) / Tpc
NSteps = Val(TxtMaxPress) / Val(txtdeltaPressure)
PathName$ = drvDrive.Drive
FLNM$ = PathName$ + "\" + txtFile1.Text
Open FLNM$ For Output As #2
Print #2, "Pressure"; ", "; "Zfactor"; ", "; "Viscosity"; ", "; "Compress"; ", "; "PseudoPressure"
With grdTable
For P = Val(txtdeltaPressure) To Val(TxtMaxPress) Step Val(txtdeltaPressure)
    .Col = 1: .row = count: .Text = P
    press(count) = P
    Ppr = P / Ppc

THIS SUBROUTIN CALCULATES THE Z-FACTOR
D5 = 0.27 * (Ppr / Tpr)
D = 0
Do While (Abs(D5 - D) > 0.0001)
    D = D5
    Let j1 = 1 + (A1 + A2 / Tpr + A3 / Tpr ^ 3 + A4 / Tpr ^ 4 + A5 / Tpr ^ 5) * D
        j2 = (A6 + A7 / Tpr + A8 / Tpr ^ 2) * D ^ 2
        j3 = A9 * (A7 / Tpr + A8 / Tpr ^ 2) * D ^ 5
        j4 = A10 * (1 + A11 * D ^ 2) * (D ^ 2 / Tpr ^ 3) * Exp(-A11 * D ^ 2)

    Z = j1 + j2 - j3 + j4
    Let D5 = 0.27 * (Ppr / (Z * Tpr))
Loop
.Col = 2: .row = count: .Text = Str(Format(Z, "0.00000")): Zfa(count) = .Text

To calculate the gas compressibility
Let G1 = A1 + A2 / Tpr + A3 / Tpr ^ 3 + A4 / Tpr ^ 4 + A5 / Tpr ^ 5
    G2 = 2 * D * (A6 + A7 / Tpr + A8 / Tpr ^ 2) - 5 * D ^ 4 * A9 * (A7 / Tpr + A8 / Tpr ^ 2)
    G3 = 2 * A10 * D / Tpr ^ 3 * (1 + A11 * D ^ 2 - A11 ^ 2 * D ^ 4) * Exp(-A11 * D ^ 2)
    G4 = G1 + G2 + G3
    Cpr = 1 / Ppr - 0.27 / (Z ^ 2 * Tpr) * (G4 / (1 + (D / Z) * G4))
    Compressibility = Cpr / Ppc
.Col = 3: .row = count: .Text = Str(Format(Compressibility, "0.0000000")): Comp(count) = .Text

Rem TO CALCULATE VISCOSITY AT HIGH PRESSURE
Ma = 29 * Val(TxtGasg)
E = (9.379 + 0.01607 * Ma) * (Val(TxtTemp) + 460) ^ 1.5 / (209.2 + 19.26 * Ma + (Val(TxtTemp) + 460))
b = 3.448 + 986.4 / (Val(TxtTemp) + 460) + 0.01009 * Ma
C = 2.447 - 0.2224 * b
Density = 2.703 * Val(TxtGasg) * P / (Z * (460 + Val(TxtTemp)))

convert density from lb/cf to g/cm^3
Density = Density * 0.016018
Viscosity = E * Exp(b * Density ^ C) * 10 ^ -4
.Col = 4: .row = count: .Text = Str(Format(Viscosity, "0.000000")): Visc(count) = .Text
.Col = 5: .row = count: .Text = Str(Format(Viscosity * Compressibility, "0.000000"))

Pseudo-Pressure calculations
tmp1 = 2 * P / (Viscosity * Z)
tmp2 = (tmp1 + Tmp) / 2
tmp3 = tmp2 * Val(txtdeltaPressure)
tmp4 = tmp3 + PseudoPressure
Tmp = tmp1
PseudoPressure = tmp4
.Col = 6: .Text = Str(PseudoPressure): PseudoP(count) = .Text
Print #2, press(count); ", "; Zfa(count); ", "; Visc(count); ", "; Comp(count); ", "; PseudoP(count) ' - 3.994 * 100000000#
count = count + 1

```



```

Next P
Close #2
End With
Picture1.ScaleHeight = 11000: Picture1.ScaleWidth = Val(TxtMaxPress) * 10
deltax = Val(TxtMaxPress) * 10 / 5
deltay = 10000 / 5
Picture1.DrawStyle = 2
For i = 1 To 5
    Picture1.Line (0, deltax * i)-(Val(TxtMaxPress) * 10, deltax * i), RGB(100, 100, 100)
    Picture1.Line (deltax * i, 0)-(deltax * i, 11000), RGB(100, 100, 100)
Next
Picture1.DrawStyle = 0
For i = 1 To count - 2
    Picture1.Line (press(i) * 10, 11000 - Zfa(i) * 11000)-(press(i + 1) * 10, 11000 - Zfa(i + 1) * 11000), RGB(225, 0, 0)
Next
txtMpres.Text = TxtMaxPress + " Psi"
End Sub

```

```

Private Sub Form_Load()
    Dim i As Integer
    With frmGasProperties
        .Left = 0: .Top = 0: .Width = 20000: .Height = 18000
    Picture1.Width = 11000: Picture1.Height = 5500
    End With
    With grdTable
        .row = 0: .Col = 1: .Text = "Pressure":
        .Col = 2: .Text = "Z-Factor"
        .Col = 4: .Text = "Viscosity"
        .Col = 3: .Text = "Compressibility"
        .Col = 5: .Text = "Comp*Visc"
        .Col = 6: .Text = "Pseudo-Pressure"
        .Col = 0
    For i = 1 To 500
        .row = i: .Text = i
    Next i
    End With
End Sub

```

Option Explicit
'variable must be declared

```

Private Sub ChkC7_Click()
    If ChkC7 = 1 Then
        lblC7.Visible = True
        txtC7.Visible = True
    Else
        lblC7.Visible = False
        txtC7.Visible = False
    End If
End Sub

```

```

Private Sub cmdClose_Click()
    frmGasComposition.Hide
    frmGasProperties.Show
End Sub

```

```

Private Sub cmdOK_Click()
    Dim y1, Y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, y13, y14, y15 As Single
    Dim X, Y, v, Ma, b5, b2, b3, Ppc1, c1, c2, c3, Tpc1, k1, Ppc7, Rgc7 As Single
    Dim E, Mc7, Yc7, Tb, Xx1, Xx2, Tpc7, W1, W2, W3, W4 As Single
    Dim title1, title2, title3, file As String
    Tpc = 0: Ppc = 0
    y1 = Val(txtn2): Y2 = Val(txtco2): y3 = Val(txth2s): y4 = Val(txtch4)

```

```

y5 = Val(TxtC2h5): y6 = Val(Txtc3h8): y7 = Val(Txtic4h10)
y8 = Val(txtnc4h10): y9 = Val(txtnc5h12): y10 = Val(txtic5h12): y11 = Val(Txtnc6h14)
y12 = Val(Txtnc7h16): y13 = Val(txtnc8h18): y14 = Val(Txtnc9h20): y15 = Val(txtnc10h22)
file = "A:\component.txt"
Open file For Input As #1
Input #1, title1, title2, title3
i = 1
Do While Not EOF(1)
Input #1, T(i), Pa(i), M(i)
i = i + 1
Loop
Close #1
X = (y1 * M(1)) + (Y2 * M(2)) + (y3 * M(3)) + (y4 * M(4)) + (y5 * M(5))
Y = (y6 * M(6)) + (y7 * M(7)) + (y8 * M(8)) + (y9 * M(9)) + (y10 * M(10))
v = (y11 * M(11)) + (y12 * M(12)) + (y13 * M(13)) + (y14 * M(14)) + (y15 * M(15))
Ma = X + Y + v
b5 = (y1 * Pa(1)) + (Y2 * Pa(2)) + (y3 * Pa(3)) + (y4 * Pa(4)) + (y5 * Pa(5))
b2 = (y6 * Pa(6)) + (y7 * Pa(7)) + (y8 * Pa(8)) + (y9 * Pa(9)) + (y10 * Pa(10))
b3 = (y11 * Pa(11)) + (y12 * Pa(12)) + (y13 * Pa(13)) + (y14 * Pa(14)) + (y15 * Pa(15))
Ppc1 = b5 + b2 + b3
c1 = (y1 * T(1)) + (Y2 * T(2)) + (y3 * T(3)) + (y4 * T(4)) + (y5 * T(5))
c2 = (y6 * T(6)) + (y7 * T(7)) + (y8 * T(8)) + (y9 * T(9)) + (y10 * T(10))
c3 = (y11 * T(11)) + (y12 * T(12)) + (y13 * T(13)) + (y14 * T(14)) + (y15 * T(15))
Tpc1 = c1 + c2 + c3

'TO CORRECT FOR IMPURITIES OF H2S AND CO2
k1 = Y2 + y3
E = 120 * (k1 ^ (0.9) - k1 ^ (1.6)) + 15 * (y3 ^ (0.5) - y3 ^ (4))

' THE CORRECTED (ADJUSTED) PSEUDOCRITICAL TEMPRETURE AND PRESSURE
Tpc = Tpc1 - E
Ppc = (Ppc1 * Tpc) / (Tpc1 + y3 * (1 - y3) * E)

' THE MOLECULAR FRACTRUE OF C7+"; YC7
If Chkc7 = 1 Then
Yc7 = Val(txtC7)
Mc7 = 114.23
Tb = (4.5579 * (Mc7 ^ (0.15178)) * (Yc7 ^ (0.15427))) ^ 3
Xx1 = 341.7 + 811 * Yc7 + (0.4244 + 0.1174 * Yc7) * Tb
Xx2 = (0.4669 - 3.2623 * Yc7) * 10 ^ 5 / (Tb ^ 2)
Tpc7 = Xx1 + Xx2
W1 = (8.3634 - 0.0566 / Yc7)
W2 = (0.24244 + 2.2898 / Yc7 + 0.11857 / Yc7 ^ 2) * 10 ^ -3 * Tb
W3 = (1.4685 + 3.684 / Yc7 + 0.4227 / Yc7 ^ 2) * 10 ^ -7 * Tb ^ 2
W4 = (0.42019 + 1.6977 / Yc7 ^ 2) * 10 ^ -10 * Tb ^ 3
Ppc7 = Exp(W1 - W2 + W3 - W4)
Rgc7 = (Yc7 * Mc7) / 29
Else
Tpc7 = 0: Ppc7 = 0: Yc7 = 0
End If

'THE FINAL CORRECTED PSEDOCRITICAL TEMP AND PRESSURE AFTER THE
'EFFECT OF THE C7+ IS
Tpc = Tpc + Tpc7
Ppc = Ppc + Ppc7

'THE SPECIFIC GRAVITY WILL BE
Gasg = (Ma + (Yc7 * Mc7)) / 29
gflag = 1
With frmGasProperties
.lblPpc.Visible = True: .lbltpc.Visible = True
.txtGasg = Gasg: .txtTpc.Text = Tpc
.txtTpc.Visible = True: .txtPpc.Text = Ppc
.txtPpc.Visible = True: .Show
End With
End Sub

Option Explicit
'all variables must be declared

```

```

Private Sub chkYes_Click()
If chkYes.Value = 1 Then
    frmReservoir.Visible = True
Else
    frmReservoir.Visible = False
End If
End Sub

Private Sub cmdBack_Click()
frmDeclineCurve.Hide
frmGasProperties.Show
End Sub

Private Sub cmdExit_Click()
End
End Sub

Private Sub cmdGraph_Click()
frmDimensionless.Show
End Sub

Private Sub cmdPrint1_Click()
frmDeclineCurve.PrintForm
End Sub

Private Sub cmdOK1_Click()
Dim Max, Min, avgD As Single
N3 = Val(txtstp)
Max = Val(txtMaxValue)
Min = Val(txtMinValue)
avgD = (Max - Min) / N3
If optperm(1).Value = True Then
    update
    For i = 1 To N3
        K(i) = i * avgD
    Next
ElseIf OptPor(2).Value = True Then
    update
    For i = 1 To N3
        porosity(i) = i * avgD
    Next
ElseIf optSkin(3).Value = True Then
    update
    For i = 1 To N3
        S(i) = i * avgD
    Next
ElseIf OptShape(4).Value = True Then
    update
    For i = 1 To N3
        CA(i) = i * avgD
    Next
ElseIf OptArea(5).Value = True Then
    update
    For i = 1 To N3
        Area(i) = i * avgD
    Next
ElseIf OptRW(6).Value = True Then
    update
    For i = 1 To N3
        Rw(i) = i * avgD
    Next
ElseIf OptGasS(7).Value = True Then
    update
    For i = 1 To N3
        Sg(i) = i * avgD
    Next
ElseIf optThick(8).Value = True Then
    update
    For i = 1 To N3

```

```

        H(i) = i * avgD
    Next
ElseIf OptPflow(9).Value = True Then
    update
    For i = 1 To N3
        Pwf(i) = i * avgD
    Next
End If
lblready.Visible = True
End Sub

Public Sub cmdRun1_Click()
    Dim count, Xfxe, count4, Low, AGip, ATD, TDT, X, Y As Single
    Dim A4, B4, CAI, S1, Pwf1, DeltaPressure, ZMaxP, volume As Single
    XSpan = 0: CumYears = 0: maxTime = 0
    CumGpScf = 0: MaxProd = 0: maxQavg = 0

    'detect error of input values
    If (txtK = "" Or txtPorosity = "" Or txtArea = "" Or txtSkinfactor = "") Then
        MsgBox ("Please, Make sure all reservoir properties are inputed")
        frmDeclineCurve.Show
        Exit Sub
    End If
    If (txtShapefactor = "" Or txtRw = "" Or txtPwf = "" Or txtSg = "" Or txtH = "") Then
        MsgBox ("Please, Make sure all reservoir properties are inputed")
        frmDeclineCurve.Show
        Exit Sub
    End If

    'initializing values
    If chkYes.Value = 0 Then
        N1 = 1
        porosity(1) = Val(txtPorosity)
        K(1) = Val(txtK)
        H(1) = Val(txtH)
        CA(1) = Val(txtShapefactor)
        Area(1) = Val(txtArea)
        Pwf(1) = Val(txtPwf)
        Rw(1) = Val(txtRw)
        Sg(1) = Val(txtSg)
        S(1) = Val(txtSkinfactor)
    Else
        N1 = N3
    End If
    For j = 1 To N1
        XSpan = 0: CumYears = 0: maxTime = 0
        CumGpScf = 0: MaxProd = 0: maxQavg = 0
        DT = 0: DGp = 0: Dq = 0: count = 0
        If j = 1 Then

            'open the gas composition file for read
            PathName = frmGasProperties!drvDrive.Drive
            GasComResult = PathName + "\" + frmGasProperties!txtFile1.Text
            Open GasComResult For Input As #1
            Input #1, tit1, tit2, tit3, tit4, tit5
            i = 1
            Do While Not EOF(1)
                Input #1, PRead(i), ZRead(i), VRead(i), chart2, MPRead(i)
                i = i + 1
            Loop
            Close #1

            'refresh the input values
            DeltaPressure = Val(frmGasProperties!txtdeltaPressure.Text)
            MaxP = Val(frmGasProperties!TxtMaxPress.Text)
            Temp = Val(frmGasProperties!TxtTemp.Text)
            G = Val(frmGasProperties!TxtGasg.Text)
            Xfxe = Val(txtfrac)
        End If
    Next j
End Sub

```

```

' Initializing values
Qi = 0
Zi = ZRead(1)
Pi = MaxP
Re = Sqr(Area(j) * 43560 / 3.1415)
If (Chkbackpressure.Value = 0) Then
    Low = Pwf(j)
Else
    Low = 700
End If

If j = 1 Then
    pathname1$ = frmDeclineCurve.drvDrive1.Drive
    FINM1$ = pathname1$ + "\" + txtDvalues.Text
    Open FINM1 For Output As #2
    Print #2, "Pressure"; ", "; "Td"; ", "; "Qd"; ", "; "GPd"; ", "; "CumDays"; ", "; "Qavg"; ", "; "CumGPscf"
End If

'To find the number of steps
R1 = MaxP / DeltaPressure
With grdDecline
For i = 0 To 5000
    count = count + 1
    Pnow = MaxP - i * DeltaPressure
    .Col = 1: .row = i + 1: .Text = Pnow

'Reservoir Properties (R)(starting from the Maximum pressure which is equal to initial reservoir pressure)
'read the pressure and z from the file
R = Pnow / DeltaPressure
Pnow = PRead(R)
Znow = ZRead(R)
.Col = 2: .row = i + 1: .Text = Format(Znow, "0.0000")
MPR = MPRead(R)
.Col = 3: .row = i + 1: .Text = Format(MPR, "#####.0")
ViscR = VRead(R)
.Col = 4: .row = i + 1: .Text = Format(ViscR, "0.000000")

'Flowing well properties (WF)
If (Chkbackpressure = 1) Then
    Pwf(j) = Pwf(j) * Val(txtfraction)
End If
R = Pwf(j) / DeltaPressure
P1 = PRead(R)
Z1 = ZRead(R)
MP1 = MPRead(R)
Visc1 = VRead(R)
P2 = PRead(R + 1)
Z2 = ZRead(R + 1)
MP2 = MPRead(R + 1)
Visc2 = VRead(R + 1)

'Call a function for interpolation
Interpolation

' Z-factor of the Max. Pressure
ZMaxP = ZRead(R1)

'Calculation of flow Rate (Q) from Deliverability Equation
ViscAvg = (ViscR + ViscWF) / 2
DeltaMP = MPR - MPWF
Area(j) = 3.1415 * (Re) ^ 2
Xi(j) = (MaxP / ZRead(R1) / (Pwf(j) / Zwf(j)))
Area1 = Area(j) / 43560#

'Check if well is fractured or not
If (Chkfracture.Value = 1) Then
    Term = (10.06 * Xfxe ^ 2)
Else
    Term = (10.06 * Area(j)) / (CA(j) * (Rw(j)) ^ 2)
End If

```

```

' continue on the equations
a = (1422 * (Temp + 460)) / (K(j) * H(j)) * ((0.5 * Log(Term)) - 0.75 + S(j))
Beta = (27300000000#) / ((K(j)) ^ 1.1045)
b = ((0.00000000003161) * Beta * (Temp + 460) * G) / (H(j) ^ 2 * ViscAvg) * (1 / Rw(j) - 1 / Re)
MQ = Sqr(a ^ 2 + 4# * b * DeltaMP)
Q = (-a + MQ) / (2# * b)
Qnow = Q
If (count = 1) Then
    QiQ = Q * 1.006
    Fndi = 1 + b / a * QiQ
End If
If (Qi = 0) Then GoTo 5000
If (count4 = 1) Then
    count4 = 2
    GoTo 5000
End If
Qavg = (Qnow + Qi) / 2

'Material Balance Equation
Bgi = 0.02829 * Zi * (Temp + 460#) / Pi
Bgf = 0.02829 * Znow * (Temp + 460#) / Pnow
Gi = Area(j) * H(j) * porosity(j) * Sg(j) * (1 / Bgi)
If (count = 2) Then
    iGp = Gi
End If

'second well option or when Xi ration is variable or stimulation occurs
If (Chkchanging.Value = 1) Then
    A4 = Val(TxtfractionGi.Text)
    B4 = Val(txtfractionRe.Text)
    CAI = Val(txtshapefact.Text)
    S1 = Val(txtSkinfact.Text)
    Pwf1 = Val(txtPwfchange.Text)
    AGip = iGp * (1 - A4)
    If (Gi <= AGip) Then count2 = count2 + 1
    If (chkSpacing.Value = 1 And count2 = 1) Then
        Re = Re * Sqr(B4)
        CA(j) = CAI
    End If
    If (ChkSkinfact.Value = 1 And Gi <= AGip) Then
        S(j) = S1
    End If
    If (ChkPwf.Value = 1 And Gi <= AGip) Then
        count4 = count4 + 1
        Pwf(j) = Pwf1
        Low = Pwf1
    End If
End If
GpScf = (Bgf - Bgi) / Bgf
GpScf = Gi * GpScf
GpMcf = GpScf / 1000#
CumGpScf = CumGpScf + GpScf
TDays = GpMcf / Qavg
Tyears = TDays / 365#
CumYears = CumYears + Tyears
Dq = Qavg / QiQ
DGp = CumGpScf / iGp
DT = (QiQ * CumYears * 365) / iGp * 1000

'This section Produce the value for deltaT
If (Gi = AGip) Then count3 = count3 + 1
If (count3 = 1) Then ATD = DT
If (count3 >= 1) Then
    TDT = DT
    DTD = Abs(TDT - ATD)
End If

'end of deltaT
CumDays = CumYears * 365

```

```

If DeltaMP <> 0 Then
    Z9 = (b * Qavg ^ 2) / DeltaMP * 100
End If
X1(j, i) = DT
.Col = 5: .row = i + 1: .Text = Format(DT, "###0.0000")
X2(j, i) = Dq
.Col = 6: .row = i + 1: .Text = Format(Dq, "0.000000")
X3(j, i) = Qavg
.Col = 7: .row = i + 1: .Text = Format(Qavg, "##0.00")
X4(j, i) = DGp
.Col = 8: .row = i + 1: .Text = Format(DGp, "##0.000000")
X5(j, i) = CumGpScf
.Col = 9: .row = i + 1: .Text = Format(CumGpScf, "##.")

' change the cumulative production from scf to Mcf
X5a(j, i) = CumGpScf / 1000
X6(j, i) = CumDays
.Col = 10: .row = i + 1: .Text = Format(CumDays, "##0.0")
X7(j, i) = DTD
.Col = 11: .row = i + 1: .Text = DTD
Print #2, Pnow; ", ", Format(X1(j, i), "0.0000"); ", ", Format(X2(j, i), "0.0000"); ", ", Format(X4(j, i), "0.000000"); ", ", Format(X6(j,
i), "0.00"); ", ", Format(X3(j, i), "0.00"); ", ", Format(X5(j, i), "0.00")
If (DT < 0 Or Dq < 0 Or DGp < 0) Then
    Exit For
End If
5000 Qi = Qnow
Zi = Znow
Pi = Pnow
If ((Pnow - DeltaPressure) < Low) Then
    Exit For
End If
XSpan = XSpan + 1
Next i
End With
If j = N1 Then
    Exit For
End If
Print #2, ""
Next j
If frmDeclineCurve.chkYes.Value = 0 Then
Print #2, ""
Print #2, ""
Print #2, "Input Data", "", "Output Data"
Print #2, ""
Print #2, "Porosity ="; porosity(1), "Xi = "; Format(Xi(1), "0.00")
Print #2, "K = "; K(1); " md", "", "Qi = "; Format(QiQ, "0.00"); " Mcf"
Print #2, "H = "; H(1); "ft", "", "Fndi = "; Format(Fndi, "0.00")
Print #2, "CA = "; CA(1), "", "Gi = "; Format(iGp / 10 ^ 9, "0.00000"); " Bcf"
Print #2, "Area = "; Area1; "acres"
Print #2, "Pi = "; MaxP; "psi"
Print #2, "Pwf = "; Pwf(1); "psi"
Print #2, "Rw = "; Rw(1); "ft"
Print #2, "Sg = "; Sg(1)
Print #2, "S = "; S(1)
End If
Close #2
lblready.Visible = False
Graph
End Sub

Private Sub Form_Load()
Dim i As Integer
With frmDeclineCurve
.Top = 0: .Left = 0: .Width = 20000: .Height = 20000
Picture1.Width = 7000: Picture1.Height = 6000
Picture2.Width = 7000: Picture2.Height = 6000
End With
'initializing Decline Curve Table
With grdDecline

```

```

.row = 0
.Col = 1: .Text = "Pressure": .Col = 2: .Text = "Zfactor"
.Col = 3: .Text = "PseudoPressure": .Col = 4: .Text = "Viscosity"
.Col = 5: .Text = "DTime": .Col = 6: .Text = "DFlowRate"
.Col = 7: .Text = "Ave.FlowR": .Col = 8: .Text = "DGasProd"
.Col = 9: .Text = "CumProd": .Col = 10: .Text = "CumDays"
.Col = 11: .Text = "DTD"
For i = 1 To 500
.Col = 0: .row = i: .Text = i
Next
End With
End Sub

```

```

Public Sub Interpolation()
Diff = P2 - P1
Diff1 = Pwf(j) - P1
ratio = Diff1 / Diff
Dif = Z1 - Z2
Zwf(j) = Z1 - (ratio * Dif)
Dif = MP2 - MP1
MPWF = MP1 + (ratio * Dif)
Dif = Visc2 - Visc1
ViscWF = Visc1 + (ratio * Dif)
End Sub

```

```

Private Sub chkbackpressure_Click()
If (Chkbackpressure.Value = 1) Then
lblReservoirfracture.Visible = True
txtfraction.Visible = True
Else
lblReservoirfracture.Visible = False
txtfraction.Visible = False
End If
End Sub

```

```

Private Sub chkchanging_Click()
If (Chkchanging.Value = 1) Then
lblfractionGi.Visible = True
TxtfractionGi.Visible = True
ChkSkinfact.Visible = True
ChkPwf.Visible = True
chkSpacing.Visible = True
Else
lblfractionGi.Visible = False
TxtfractionGi.Visible = False
ChkSkinfact.Visible = False
ChkPwf.Visible = False
chkSpacing.Visible = False
txtSkinfact.Visible = False
txtshapefact.Visible = False
txtPwfchange.Visible = False
txtfractionRe.Visible = False
lblfractionRe.Visible = False
lblshapefact.Visible = False
End If
End Sub

```

```

Private Sub chkfracture_Click()
If Chkfracture.Value = 1 Then
lblFrac.Visible = True
txtfrac.Visible = True
Else
lblFrac.Visible = False
txtfrac.Visible = False
End If
End Sub

```



```

Private Sub chkPwf_Click()
If ChkPwf.Value = 1 Then
    txtPwfchange.Visible = True
Else
    txtPwfchange.Visible = False
End If
End Sub

```

```

Private Sub chkspacing_Click()
If chkSpacing.Value = 1 Then
    txtfractionRe.Visible = True
    lblfractionRe.Visible = True
    txtshapefact.Visible = True
    lblshapefact.Visible = True
Else
    txtfractionRe.Visible = False
    lblfractionRe.Visible = False
    txtshapefact.Visible = False
    lblshapefact.Visible = False
End If
End Sub

```

```

Private Sub chkSkinfact_Click()
If ChkSkinfact.Value = 1 Then
    txtSkinfact.Visible = True
Else
    txtSkinfact.Visible = False
End If
End Sub

```

```

Public Sub Graph()
Dim X, Y, flag As Single
flag = 0
Picture1.Cls
Picture2.Cls
For j = 1 To N1
For i = 1 To XSpan
    If (X6(j, i) > maxTime) Then maxTime = X6(j, i)
    If (X5a(j, i) > MaxProd) Then MaxProd = X5a(j, i)
    If (X3(j, i) > maxQavg) Then maxQavg = X3(j, i)
Next
Next

' if time is big change it to years
If (maxTime > 1000000) Then
    maxTime = maxTime / 365
    delX = maxTime / 4
    flag = 1
End If
Picture1.ScaleWidth = maxTime
Picture2.ScaleWidth = maxTime
Picture1.ScaleHeight = MaxProd
Picture2.ScaleHeight = maxQavg
If (maxTime <= 100) Then delX = 10
If (maxTime <= 1000) Then delX = 100
If (maxTime <= 10000) Then delX = 1000
Picture1.DrawStyle = 2: Picture2.DrawStyle = 2
For i = 1 To XSpan + 1
    Picture1.Line (i * delX, 0)-(i * delX, MaxProd), RGB(125, 125, 25)
    Picture2.Line (i * delX, 0)-(i * delX, maxQavg), RGB(125, 125, 25)
Next
delY1 = MaxProd / 4: delY2 = maxQavg / 4
For i = 1 To 3
    Picture1.Line (0, i * delY1)-(maxTime, i * delY1), RGB(125, 125, 25)
    Picture2.Line (0, i * delY2)-(maxTime, i * delY2), RGB(125, 125, 25)

```

```

Next
For j = 1 To N1
    Picture1.DrawStyle = 0: Picture2.DrawStyle = 0
    Picture1.Line (0, MaxProd)-(X6(j, 1), MaxProd - X5(j, 1)), RGB(255, 0, 0)
For i = 1 To XSpan - 2
    Picture1.Line (X6(j, i + 1), MaxProd - X5a(j, i + 1))-(X6(j, i + 2), MaxProd - X5a(j, i + 2)), RGB(255, 0, 0)
    Picture2.Line (X6(j, i), maxQavg - X3(j, i))-(X6(j, i + 1), maxQavg - X3(j, i + 1)), RGB(255, 0, 0)
Next i
If j = N1 Then Exit For
Next j
If flag = 0 Then
    txtmaxTime2.Text = Format(maxTime, "0.0") + " days"
    txtmaxtime1.Text = Format(maxTime, "0.0") + " days"
Else
    txtmaxTime2.Text = Format(maxTime, "0.0") + " years"
    txtmaxtime1.Text = Format(maxTime, "0.0") + " years"
End If
txtmaxGp.Text = Format(MaxProd / 1000000, "0.000") + " Bcf"
txtmaxQavg.Text = Format(maxQavg, "0.000") + " Mcfd"
End Sub

Public Sub update()
    For i = 1 To N3
        porosity(i) = Val(txtPorosity)
        K(i) = Val(txtK)
        H(i) = Val(txtH)
        CA(i) = Val(txtShapefactor)
        Area(i) = Val(txtArea)
        Pwf(i) = Val(txtPwf)
        Rw(i) = Val(txtRw)
        Sg(i) = Val(txtSg)
        S(i) = Val(txtSkinfactor)
    Next
End Sub

'all variables must be declared
Option Explicit

Private Sub cmdBack2_Click()
    frmDeclineCurve.Show
End Sub

Private Sub cmdExit3_Click()
    End
End Sub

Private Sub cmdGraph_Click()
    Dim column, row, ki, sam As Integer
    Dim PathName2, DeclineResult As String
    ki = 1: j = 1
    If N3 = 0 Then N3 = 1
    With frmDimensionless.MSChart2
        .chartType = VtChChartType2dXY
        .ColumnCount = N3 * 2
        .RowCount = XSpan

        For ki = 1 To N3 * 2
            For row = 1 To XSpan
                .column = ki
                .row = row
                If (ki = 1 Or ki = 3 Or ki = 5 Or ki = 7 Or ki = 9 Or ki = 11 Or ki = 13 Or ki = 15 Or ki = 17 Or ki = 19) Then
                    .Data = X1((ki + 1) / 2, row)
                Else
                    .Data = X2(ki / 2, row)
                End If
            Next
        Next
    End With

```

```

End With
With frmDimensionless.MSChart1
    .chartType = VtChChartType2dXY
    .ColumnCount = N3 * 2
    .RowCount = XSpan
    For ki = 1 To N3 * 2
        For row = 1 To XSpan
            .column = ki
            .row = row
            If (ki = 1 Or ki = 3 Or ki = 5 Or ki = 7 Or ki = 9 Or ki = 11 Or ki = 13 Or ki = 15 Or ki = 17 Or ki = 19) Then
                .Data = X1((ki + 1) / 2, row)
            Else
                .Data = X4(ki / 2, row)
            End If
        Next
    Next
End With
txtGi.Text = Format(iGp / 1000000000, "0.00000"):    txtXi.Text = Format(Xi(j), "0.00")
txtFndi.Text = Format(Fndi, "0.00"):                txtQi.Text = Format(QiQ, "0.00")
txtA.Text = Format(a, "0.00"):                      txtB.Text = Format(b, "0.00")

' used as input for CurveMatch computer Program
Open "a:Original.txt" For Output As #7
Print #7, "Time", "Qavg", "Gp(Mcf)"
For sam = 1 To XSpan
    Print #7, Format(X6(1, sam), "0.00"), Format(X3(1, sam), "0.00"), Format(X5a(1, sam), "0.00")
Next
Close #7

' Used as input for Type Curve Generator
Open "a:Td.txt" For Output As #8
Print #8, "Td"
For sam = 1 To XSpan
    Print #8, Format(X1(1, sam), "0.000000")
Next
Close #8
End Sub

Private Sub cmdZoomout_Click()
    frmGraph1.Plot1
    frmGraph1.Show
End Sub

Private Sub cmdZoomout2_Click()
    frmGraph2.Plot2
    frmGraph2.Show
End Sub

Private Sub Form_Load()
    frmDimensionless.Top = 0:    frmDimensionless.Left = 0
    frmDimensionless.Width = 19000:    frmDimensionless.Height = 10000
    With frmDimensionless.MSChart1.Plot.Axis(VtChAxisIdX)
        .AxisScale.Type = VtChScaleTypeLogarithmic
        .AxisGrid.MinorPen.Style = VtPenStyleNull
        .CategoryScale.Auto = False
        .ValueScale.Maximum = 100
        .ValueScale.Minimum = 0.001
    End With
    With frmDimensionless.MSChart1.Plot.Axis(VtChAxisIdY)
        .AxisScale.Type = VtChScaleTypeLogarithmic
        .AxisGrid.MinorPen.Style = VtPenStyleNull
        .CategoryScale.Auto = False
        .ValueScale.Maximum = 100
        .ValueScale.Minimum = 0.0001
    End With
    With frmDimensionless.MSChart2.Plot.Axis(VtChAxisIdX)

```

```

        .AxisScale.Type = VtChScaleTypeLogarithmic
        .AxisGrid.MinorPen.Style = VtPenStyleNull
        .CategoryScale.Auto = False
        .ValueScale.Maximum = 100
        .ValueScale.Minimum = 0.001
    End With
    With frmDimensionless.MSChart2.Plot.Axis(VtChAxisIdY)
        .AxisScale.Type = VtChScaleTypeLogarithmic
        .AxisGrid.MinorPen.Style = VtPenStyleNull
        .CategoryScale.Auto = False
        .ValueScale.Maximum = 100
        .ValueScale.Minimum = 0.0001
    End With
End Sub

```

```

' all variables must be declared
Option Explicit

```

```

Private Sub Form_DblClick()
    frmGraph1.Hide
End Sub

```

```

Private Sub Form_Load()
    With frmGraph1.MSChart1.Plot.Axis(VtChAxisIdX)
        .AxisScale.Type = VtChScaleTypeLogarithmic
        .AxisGrid.MinorPen.Style = VtPenStyleNull
        .CategoryScale.Auto = False
        .ValueScale.Maximum = 100
        .ValueScale.Minimum = 0.001
    End With
    With frmGraph1.MSChart1.Plot.Axis(VtChAxisIdY)
        .AxisScale.Type = VtChScaleTypeLogarithmic
        .AxisGrid.MinorPen.Style = VtPenStyleNull
        .CategoryScale.Auto = False
        .ValueScale.Maximum = 100
        .ValueScale.Minimum = 0.0001
    End With
End Sub

```

```

Public Sub Plot1()
    Dim k1 As Integer
    With frmGraph1.MSChart1
        .chartType = VtChChartType2dXY
        .ColumnCount = N3 * 2
        .RowCount = XSpan
        For ki = 1 To N3 * 2
            For row = 1 To XSpan
                .column = ki
                .row = row
                If (ki = 1 Or ki = 3 Or ki = 5 Or ki = 7 Or ki = 9 Or ki = 11 Or ki = 13 Or ki = 15 Or ki = 17 Or ki = 19) Then
                    .Data = X1((ki + 1) / 2, row)
                Else
                    .Data = X2(ki / 2, row)
                End If
            Next
        Next
    End With
End Sub

```

```

'All variables must be declared
Option Explicit
Private Sub Form_DblClick()
    frmGraph2.Hide
End Sub

```

```

Private Sub Form_Load()
With frmGraph2.MSChart1.Plot.Axis(VtChAxisIdX)
    .AxisScale.Type = VtChScaleTypeLogarithmic
    .AxisGrid.MinorPen.Style = VtPenStyleNull
    .CategoryScale.Auto = False
    .ValueScale.Maximum = 100
    .ValueScale.Minimum = 0.001
End With
With frmGraph2.MSChart1.Plot.Axis(VtChAxisIdY)
    .AxisScale.Type = VtChScaleTypeLogarithmic
    .AxisGrid.MinorPen.Style = VtPenStyleNull
    .CategoryScale.Auto = False
    .ValueScale.Maximum = 100
    .ValueScale.Minimum = 0.0001
End With
End Sub

```

```

Public Sub Plot2()
Dim k1 As Integer
With frmGraph2.MSChart1
    .chartType = VtChChartType2dXY
    .ColumnCount = N3 * 2
    .RowCount = XSpan
    For ki = 1 To N3 * 2
        For row = 1 To XSpan
            .column = ki
            .row = row
            If (ki = 1 Or ki = 3 Or ki = 5 Or ki = 7 Or ki = 9 Or ki = 11 Or ki = 13 Or ki = 15 Or ki = 17 Or ki = 19) Then
                .Data = X1((ki + 1) / 2, row)
            Else
                .Data = X4(ki / 2, row)
            End If
        Next
    Next
End With
End Sub

```

APPENDIX C

LIST MATCHCURVE

'all variables must be declared

Option Explicit

Global GasComResult, tit1, tit2, tit3, tit4, tit5, tit6, tit7, Say, chart2 As String
Global Zwf, Xi, MaxP, Temp, G, DeltaP, Pi, Pnow, Z, Pwf As Single
Global PRead(1001), ZRead(1001), MPRead(1001), VRead(1001) As Single
Global M(20), Pa(20), T(20), Gi1(20), A(20), Fndif(20), Fndii(20), Fndi(20), Tp1(20, 1001), Qp1(20, 1001) As Single
Global R, theindex, theindex1, Ndata, NSteps, IterateGi, matchGi, matchA As Integer
Global i, j, N1, N3, Check1, R1, gflag, ZGi, ZA, k, countA, countG, final, final2 As Integer
Global Znow, MPR, ViscR, ViscWF, ViscAvg, MPWF, DeltaMP, DeltaMPi As Single
Global b, B1, MQ, Q, Qnow, Qi, Qavg, Bgi, Bgf, Gi, Zi As Single
Global GpScf, GpMcf, CumGpScf, TDays, Tyears, CumYears, CumDays
Global XSpan, iGp, QiQ, Time(20, 1001), Qorg(20, 1001), Gporg(20, 1001) As Single
Global Gp1Mcf(10, 1001), Qn(10, 1001), RPressure(10, 1001), RDT(10, 1001), Qp(10, 1001), Gp(10, 1001), Tp(10, 1001) As Single
Global Tf(10, 1001), Qf(10, 1001), Gpf(10, 1001), min(11), Af(11), Bf(11), Gif(11) As Single
Global RDQ(10, 1001), RDGP(10, 1001), RQavg(10, 1001), RCumDays(10, 1001), RCumGPscf(10, 1001) as Single
Global P1, P2, Z1, Z2, MP1, MP2, Visc1, Visc2, Gasg, Tpc, Ppc As Single
Global delX, delY1, delY2, DelX1, DelDY1, DelDY2 As Long
Global MaxProd, maxQavg, maxTime, MaxGi, MinGi, MaxA, MinA, steps, steps1 as Single
Global MinlsA, MinlsGi, minls, Low, Volume(20), DeltaPressure As Single
Global PathName\$, FLNMS\$, pathname1\$, FINM1\$, GasCompositionResult As String

Dim A1, A2, A3, A4, A5, A6, j1, j2, j3, j4 As Single
Dim A7, A8, A9, A10, A11, D, E, F, C, Density As Single
Dim Ma, G1, G2, G3, G4, Cpr, count, Ppr, Tpr, D5, P As Single
Dim Zfactor, Compressibility, Viscosity As Single
Dim Tmp, tmp1, tmp2, tmp3, tmp4 As Single
Dim PseudoPressure As Long
Dim press(1001), Zfa(1001), Comp(1001), Visc(1001), PseudoP(1001) As Single

'variable must be declared

Option Explicit

Private Sub cmdExit_Click()

End

End Sub

Private Sub cmdGasComposition_Click()

frmGasCompo.Show

End Sub

Private Sub cmdGoToDeclineCurve_Click()

frmCompareC.Show

End Sub

Private Sub cmdPrint_Click()

frmGasProp.PrintForm

End Sub

Private Sub cmdRun_Click()

Dim A1, A2, A3, A4, A5, A6, j1, j2, j3, j4 As Single

Dim A7, A8, A9, A10, A11, D, E, b, C, Density As Single

Dim Gasg, Ppc, Ppr, Tpr, Tpc, D5, Z As Single

Dim Ma, G1, G2, G3, G4, Cpr, count, P As Single

Dim Zfactor, Compressibility, Viscosity As Single

Dim Tmp, tmp1, tmp2, tmp3, tmp4, DeltaPressure As Single

Dim PathName\$, FLNMS\$, GasCompositionResult As String

Dim PseudoPressure As Long

```

Dim ki, row As Integer
Dim press(1001), deltaPress(1001), Zfa(1001), Comp(1001), Visc(1001), PseudoP(1001) As Single

gflag = 0

PseudoPressure = 0
Tmp = 0
Let A1 = 0.3265: A2 = -1.07: A3 = -0.5339: A4 = 0.01569: A5 = -0.05165
A6 = 0.5475: A7 = -0.7361: A8 = 0.1844: A9 = 0.1056: A10 = 0.6134: A11 = 0.721
count = 1
If gflag = 0 Then
    Ppc = 709.604 - 58.718 * Val(TxtGasg)
    Tpc = 170.491 + 307.344 * Val(TxtGasg)
Else
    Ppc = Val(TxtPpc)
    Tpc = Val(txtTpc)
    TxtGasg.Text = Gasg
End If

Tpr = (Val(txttemp) + 460) / Tpc
NSteps = Val(txtMaxPress) / Val(txtdeltaPressure)

PathName$ = drvDrive.Drive
FLNM$ = PathName$ + "\" + txtFile1.Text
Open FLNM$ For Output As #2
Print #2, "Pressure"; ", "; "Zfactor"; ", "; "Viscosity"; ", "; "Compress"; ", "; "C*u"; ", "PseudoPressure"
With grdTable

For P = Val(txtdeltaPressure) To Val(txtMaxPress) Step Val(txtdeltaPressure)
    .Col = 1: .row = count: .Text = P: press(count) = P
    Ppr = P / Ppc
    THIS SUBROUTIN CALCULATES THE Z-FACTOR

    D5 = 0.27 * (Ppr / Tpr)
    D = 0
    Do While (Abs(D5 - D) > 0.0001)
        D = D5
        Let j1 = 1 + (A1 + A2 / Tpr + A3 / Tpr ^ 3 + A4 / Tpr ^ 4 + A5 / Tpr ^ 5) * D
        j2 = (A6 + A7 / Tpr + A8 / Tpr ^ 2) * D ^ 2
        j3 = A9 * (A7 / Tpr + A8 / Tpr ^ 2) * D ^ 5
        j4 = A10 * (1 + A11 * D ^ 2) * (D ^ 2 / Tpr ^ 3) * Exp(-A11 * D ^ 2)

        Z = j1 + j2 - j3 + j4
        Let D5 = 0.27 * (Ppr / (Z * Tpr))
    Loop

    .Col = 2: .row = count: .Text = Str(Format(Z, "0.00000")): Zfa(count) = .Text

    To calculate the gas compressibility
    Let G1 = A1 + A2 / Tpr + A3 / Tpr ^ 3 + A4 / Tpr ^ 4 + A5 / Tpr ^ 5
    G2 = 2 * D * (A6 + A7 / Tpr + A8 / Tpr ^ 2) - 5 * D ^ 4 * A9 * (A7 / Tpr + A8 / Tpr ^ 2)
    G3 = 2 * A10 * D / Tpr ^ 3 * (1 + A11 * D ^ 2 - A11 ^ 2 * D ^ 4) * Exp(-A11 * D ^ 2)
    G4 = G1 + G2 + G3
    Cpr = 1 / Ppr - 0.27 / (Z ^ 2 * Tpr) * (G4 / (1 + (D / Z) * G4))

    Compressibility = Cpr / Ppc

    .Col = 3: .row = count: .Text = Str(Format(Compressibility, "0.0000000")): Comp(count) = .Text

    TO CALCULATE VISCOSITY AT HIGH PRESSURE
    Ma = 29 * Val(TxtGasg)
    E = (9.379 + 0.01607 * Ma) * (Val(txttemp) + 460) ^ 1.5 / (209.2 + 19.26 * Ma + (Val(txttemp) + 460))
    b = 3.448 + 986.4 / (Val(txttemp) + 460) + 0.01009 * Ma
    C = 2.447 - 0.2224 * b
    Density = 2.703 * Val(TxtGasg) * P / (Z * (460 + Val(txttemp)))
    'convert density from lb/cf to g/cm^3
    Density = Density * 0.016018
    Viscosity = E * Exp(b * Density ^ C) * 10 ^ -4
    .Col = 4: .row = count: .Text = Str(Format(Viscosity, "0.000000")): Visc(count) = .Text
    .Col = 5: .row = count: .Text = Str(Format(Viscosity * Compressibility, "0.000000"))

```

```

'Pseudo-Pressure calculations
tmp1 = 2 * P / (Viscosity * Z)
tmp2 = (tmp1 + Tmp) / 2
tmp3 = tmp2 * Val(txtdeltaPressure)
tmp4 = tmp3 + PseudoPressure
Tmp = tmp1
PseudoPressure = tmp4
.Col = 6: .Text = Str(PseudoPressure): PseudoP(count) = .Text:
Print #2, press(count); ", "; Zfa(count); ", "; Visc(count); ", "; Comp(count); ", "; PseudoP(count)
count = count + 1
Next P

End With
Close #2

' Graphing Z-factor
count = count - 1
With MSChart1.Plot.Axis(VtChAxisIdX)
.CategoryScale.Auto = False
.ValueScale.Maximum = Val(txtMaxPress)
.ValueScale.Minimum = 0
End With
With MSChart1.Plot.Axis(VtChAxisIdY)
.CategoryScale.Auto = False
.ValueScale.Maximum = 1
.ValueScale.Minimum = 0.6
End With
With MSChart1
.ColumnCount = 2
.RowCount = count
For ki = 1 To 2
For row = 1 To count
.Column = ki
.row = row
If ki = 1 Then
.Data = press(row)
ElseIf ki = 2 Then: .Data = Zfa(row)
End If
Next
Next
End With
End Sub

Private Sub Form_Load()
Dim i As Integer
MSChart1.Plot.UniformAxis = False
With frmGasProp
.Left = 0: .Top = 0: .Width = 20000: .Height = 18000
End With

With grdTable
.ColWidth(0) = 450: .row = 0
.Col = 1: .Text = "Pressure"
.Col = 2: .Text = "Z-Factor"
.Col = 3: .Text = "Compressibility"
.Col = 4: .Text = "Viscosity"
.Col = 5: .Text = "Comp*Visc"
.Col = 6: .Text = "Pseudo-Pressure"
.Col = 0
For i = 1 To 500
.row = i
.Text = i
Next i
End With
End Sub

```


Option Explicit
'variable must be declared

```
Private Sub ChkC7_Click()
If Chkc7 = 1 Then
    lblC7.Visible = True
    txtC7.Visible = True
Else
    lblC7.Visible = False
    txtC7.Visible = False
End If
End Sub
```

```
Private Sub cmdClose_Click()
frmGasCompo.Hide
frmGasProp.Show
End Sub
```

```
Private Sub cmdOK_Click()
Dim y1, Y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, y13, y14, y15 As Single
Dim X, Y, v, Ma, b5, b2, b3, Ppc1, c1, c2, c3, Tpc1, k1 As Single
Dim E, Mc7, Yc7, Tb, Xx1, Xx2, Tpc7, W1, W2, W3, W4, Ppc7, Rgc7 As Single
Dim title1, title2, title3, file As String
```

```
Tpc = 0: Ppc = 0
y1 = Val(txtn2): Y2 = Val(txtco2): y3 = Val(txth2s): y4 = Val(txtch4)
y5 = Val(TxtC2h5): y6 = Val(Txtc3h8): y7 = Val(Txtic4h10)
y8 = Val(txtnc4h10): y9 = Val(txtnc5h12): y10 = Val(txtic5h12): y11 = Val(Txtnc6h14)
y12 = Val(Txtnc7h16): y13 = Val(txtnc8h18): y14 = Val(Txtnc9h20): y15 = Val(txtnc10h22)
```

```
file = "A:\component.txt"
Open file For Input As #1
Input #1, title1, title2, title3
i = 1
Do While Not EOF(1)
Input #1, T(i), Pa(i), M(i)
i = i + 1
Loop
Close #1
```

```
X = (y1 * M(1)) + (Y2 * M(2)) + (y3 * M(3)) + (y4 * M(4)) + (y5 * M(5))
Y = (y6 * M(6)) + (y7 * M(7)) + (y8 * M(8)) + (y9 * M(9)) + (y10 * M(10))
v = (y11 * M(11)) + (y12 * M(12)) + (y13 * M(13)) + (y14 * M(14)) + (y15 * M(15))
Ma = X + Y + v
```

```
b5 = (y1 * Pa(1)) + (Y2 * Pa(2)) + (y3 * Pa(3)) + (y4 * Pa(4)) + (y5 * Pa(5))
b2 = (y6 * Pa(6)) + (y7 * Pa(7)) + (y8 * Pa(8)) + (y9 * Pa(9)) + (y10 * Pa(10))
b3 = (y11 * Pa(11)) + (y12 * Pa(12)) + (y13 * Pa(13)) + (y14 * Pa(14)) + (y15 * Pa(15))
Ppc1 = b5 + b2 + b3
```

```
c1 = (y1 * T(1)) + (Y2 * T(2)) + (y3 * T(3)) + (y4 * T(4)) + (y5 * T(5))
c2 = (y6 * T(6)) + (y7 * T(7)) + (y8 * T(8)) + (y9 * T(9)) + (y10 * T(10))
c3 = (y11 * T(11)) + (y12 * T(12)) + (y13 * T(13)) + (y14 * T(14)) + (y15 * T(15))
Tpc1 = c1 + c2 + c3
```

'TO CORRECT FOR IMPURITIES OF H2S AND CO2

```
k1 = Y2 + y3
E = 120 * (k1 ^ (0.9) - k1 ^ (1.6)) + 15 * (y3 ^ (0.5) - y3 ^ (4))
```

' THE CORRECTED (ADJUSTED) PSEUDOCRITICAL TEMPRETURE AND PRESSURE

```
Tpc = Tpc1 - E
Ppc = (Ppc1 * Tpc) / (Tpc1 + y3 * (1 - y3) * E)
```

' THE MOLECULAR FRACTRUE OF C7+"; YC7

```
If Chkc7 = 1 Then
Yc7 = Val(txtC7)
```

```

Mc7 = 114.23
Tb = (4.5579 * (Mc7 ^ (0.15178)) * (Yc7 ^ (0.15427))) ^ 3
Xx1 = 341.7 + 811 * Yc7 + (0.4244 + 0.1174 * Yc7) * Tb
Xx2 = (0.4669 - 3.2623 * Yc7) * 10 ^ 5 / (Tb ^ 2)
Tpc7 = Xx1 + Xx2
W1 = (8.3634 - 0.0566 / Yc7)
W2 = (0.24244 + 2.2898 / Yc7 + 0.11857 / Yc7 ^ 2) * 10 ^ -3 * Tb
W3 = (1.4685 + 3.684 / Yc7 + 0.4227 / Yc7 ^ 2) * 10 ^ -7 * Tb ^ 2
W4 = (0.42019 + 1.6977 / Yc7 ^ 2) * 10 ^ -10 * Tb ^ 3
Ppc7 = Exp(W1 - W2 + W3 - W4)
Rgc7 = (Yc7 * Mc7) / 29
Else
Tpc7 = 0: Ppc7 = 0: Yc7 = 0
End If

'THE FINAL CORRECTED PSEUDOCRITICAL TEMP AND PRESSURE AFTER THE
EFFECT OF THE C7+ IS
    Tpc = Tpc + Tpc7
    Ppc = Ppc + Ppc7

'THE SPECIFIC GRAVITY WILL BE
Gasg = (Ma + (Yc7 * Mc7)) / 29
gflag = 1
frmGasProp.lblPpc.Visible = True
frmGasProp.lbltpc.Visible = True
frmGasProp.txtGasg = Gasg
frmGasProp.txtTpc.Text = Tpc
frmGasProp.txtTpc.Visible = True
frmGasProp.txtPpc.Text = Ppc
frmGasProp.txtPpc.Visible = True
frmGasProp.Show
End Sub

'all variables must be declared
Option Explicit

Private Sub cmdBack_Click()
frmCompareC.Hide
frmGasProp.Show
End Sub
Private Sub cmdExit_Click()
End
End Sub

Public Sub cmdRun1_Click()
ReadFromFile
FndiRange
Clean
'initializing for rerun
countA = 0: countG = 0: txtfndi.Text = 1.1

'initializing for the final results
100 ZA = 0: ZGi = 0: Check1 = 0
    maxTime = 0: MaxProd = 0: maxQavg = 0
    IterateGi = 1
    Pwf = Val(txtPwf)
    MinGi = Val(txtminGi) * 1000000000: MaxGi = Val(txtmaxGi) * 1000000000
    If MinGi > MaxGi Then
        MsgBox ("MinGi > MaxGi Incorrect Input")
        frmCompareC.Show
    End Sub
End If
steps = 10: steps1 = (MaxGi - MinGi) / steps
For i = 1 To steps
    Gi1(i) = MinGi + (i - 1) * steps1
Next
QiQ = Qorg(0, 1) * 1.006

```

```

If countA = 0 Then
'open the gas composition file for read
  PathName = frmGasProp!drvDrive.Drive
  GasComResult = PathName + "\" + frmGasProp!txtFile1.Text
  Open GasComResult For Input As #1
  Input #1, tit1, tit2, tit3, tit4, tit5, tit6
  i = 1
  Do While Not EOF(1)
    Input #1, PRead(i), ZRead(i), VRead(i), chart2, MPRead(i)
    i = i + 1
  Loop
  Close #1
End If
' refresh the input values
  DeltaPressure = Val(frmGasProp!txtdeltaPressure.Text)
  MaxP = Val(frmGasProp!txtMaxPress.Text)
  Temp = Val(frmGasProp!txttemp.Text)
  G = Val(frmGasProp!TxtGasg.Text)
  R = Pwf / DeltaPressure
  P1 = PRead(R): Z1 = ZRead(R): MP1 = MPRead(R): Visc1 = VRead(R)
  P2 = PRead(R + 1): Z2 = ZRead(R + 1): MP2 = MPRead(R + 1): Visc2 = VRead(R + 1)
Interpolation1

' Initializing values
  Qi = 0: Zi = ZRead(1): Pi = MaxP
  R1 = MaxP / DeltaPressure
  Bgi = 0.02829 * ZRead(R1) * (Temp + 460#) / MaxP
  Xi = (MaxP / ZRead(R1) / (Pwf / Zwf))
  Low = Pwf + DeltaPressure

'Display the intial findi for each calculation
txtfndii.Refresh: txtfndii.Text = Fndii(k)
WorkOnGiA
IterateGi = 1: matchA = 1

' To do another round of iteration
Check1 = 1
Automatic
GetResult
If k < 10 Then
  k = k + 1
  GoTo 100
End If
FinalResult
Graph
End Sub

Private Sub Form_Load()
Dim i As Integer
With frmCompareC
  .Top = 0: .Left = 0
  .Width = 18000: .Height = 18000
End With

'initializing Decline Curve Table
With grdDecline
  .ColWidth(0) = 450: .row = 0
  .Col = 1: .Text = "Time"
  .Col = 2: .Text = "Q(Mcfd)"
  .Col = 3: .Text = "Gp(Mcf)"
End With
With grddata
  .ColWidth(0) = 450: .row = 0
  .Col = 1: .Text = "Time"
  .Col = 2: .Text = "Q(Mcfd)"
  .Col = 3: .Text = "Gp(Mcf)"
End With
For i = 1 To 500
  grdDecline.Col = 0: grdDecline.row = i: grdDecline.Text = i

```

```

    grddata.Col = 0:    grddata.row = i:    grddata.Text = i
Next
End Sub

```

```

Public Sub Interpolation1()
Dim diff, diff1, ratio, dif As Single
diff = P2 - P1:        diff1 = Pwf - P1
ratio = diff1 / diff:   dif = Z1 - Z2
Zwf = Z1 - (ratio * dif)
dif = MP2 - MP1:        MPWF = MP1 + (ratio * dif)
dif = Visc2 - Visc1:    ViscWF = Visc1 + (ratio * dif)
End Sub

```

```

Public Sub Graph()
Dim ki, row, Xspans As Integer

```

```

' Fill the gap for the graphing purpose
For row = Ndata + 2 To XSpan
    Time(0, row) = Time(0, Ndata + 1)
    Qorg(0, row) = Qorg(0, Ndata + 1)
Next

```

```

' Scaling and graphing
MscResult.Plot.UniformAxis = False
MscResult.Legend.Location.Visible = True
With MscResult.Plot.Axis(VtChAxisIdX)
    .AxisScale.Type = VtChScaleTypeLogarithmic
    .AxisGrid.MinorPen.Style = VtPenStyleNull
    .CategoryScale.Auto = False
    .ValueScale.Maximum = 1000
    .ValueScale.Minimum = 1
End With
With MscResult.Plot.Axis(VtChAxisIdY)
    .AxisScale.Type = VtChScaleTypeLogarithmic
    .AxisGrid.MinorPen.Style = VtPenStyleNull
    .CategoryScale.Auto = False
    .ValueScale.Maximum = 100000
    .ValueScale.Minimum = 100
End With
With MscResult
    .chartType = VtChChartType2dXY
    .ColumnCount = 4
    .RowCount = XSpan
    For ki = 1 To 4
        For row = 1 To XSpan
            .Column = ki
            .row = row
            If ki = 1 Then
                .Data = Time(0, row)
            ElseIf ki = 2 Then .Data = Qorg(0, row)
            ElseIf ki = 3 Then .Data = Tf(final, row)
            ElseIf ki = 4 Then .Data = Qf(final, row)
            End If
        Next
    Next
End With
End Sub

```

```

Public Sub ReadData()
Dim cell, i1 As Integer

cell = 0
With grddata
    .ColAlignment(0) = 2
    For i1 = 1 To 200
        DoEvents
        cell = cell + 1
    Next
End With

```

```

Time(0, i1) = .TextMatrix(cell, 1)
Qorg(0, i1) = .TextMatrix(cell, 2)
If (.TextMatrix(cell, 1) = "") Then
    Exit For
End If
Next
End With
Ndata = cell - 1
End Sub

```

```

Private Sub Grddata_KeyPress(KeyAscii As Integer)
With grddata
.Text = .Text + Chr$(KeyAscii)
If KeyAscii = 8 Then
.Text = ""
Exit Sub
ElseIf KeyAscii = 32 Then
.Text = ""
Exit Sub
End If
End With
End Sub

```

```

Public Sub ReadFromFile()
Dim title2, title3 As String
Dim i5, J5 As Integer
Open "a:original.txt" For Input As #2
i5 = 0: J5 = 0
Input #2, title2
Do While Not EOF(2)
i5 = i5 + 1
Input #2, Time(J5, i5), Qorg(J5, i5), Gporg(J5, i5)
With grddata
DoEvents
.Col = 1: .row = i5
.Text = Val(Time(0, i5))
.Col = 2: .row = i5
.Text = Val(Qorg(0, i5))
.Col = 3: .Text = Val(Gporg(0, i5))
End With
Loop
Ndata = i5 - 1
Close #2
End Sub

```

```

Public Sub FinalResult()
' compare the finalists pick the best and the second best
For k = 0 To 10
If min(k) < minls Then
final2 = final
final = k
minls = min(k)
End If
If (min(final2) > min(k) And final <> k) Then final2 = k
Next
fmfinalresult.Visible = True
txtAMa.Text = Af(final): txtAMi.Text = Af(final2)
txtBMa.Text = Bf(final): txtBMi.Text = Bf(final2)
txtGiMa.Text = Gif(final): txtGiMi.Text = Gif(final2)
txtfndiMa.Text = Fndif(final): txtfndiMi.Text = Fndif(final2)
If (Af(final2) > Af(final)) Then: txtAMi.Text = Af(final): txtAMa.Text = Af(final2)
If Bf(final2) > Bf(final) Then: txtBMi.Text = Bf(final): txtBMa.Text = Bf(final2)
If Gif(final2) > Gif(final) Then: txtGiMi.Text = Gif(final): txtGiMa.Text = Gif(final2)
If Fndif(final2) > Fndif(final) Then: txtfndiMi.Text = Fndif(final): txtfndiMa.Text = Fndif(final2)

Open "c:Result.txt" For Output As #3

```

```

Print #3, "Time", "Qcal", "Time", "Qorg", "Qinterpu"

For i = 1 To XSpan
    With grdDecline
        .Col = 1: .row = i: .Text = Format(Tf(final, i), "0.0")
        .Col = 2: .row = i: .Text = Format(Qf(final, i), "0.0")
        .Col = 3: .row = i: .Text = Format(Gpf(final, i), "0.0")
    End With
    Print #3, Format(Tf(final, i), "0.00"), Format(Qf(final, i), "0.000"), Time(0, i), Qorg(0, i)
Next
Close #3
End Sub

```

```

Public Sub leastsquare()
Dim Bx, D(20, 1001), ls1(20), Differ(20) As Single
Differ(1) = 0: minls = 1000000000
For j = 1 To steps
    Differ(j) = 0
    For i = 1 To Ndata
        D(j, i) = Abs(Qn(j, i) - Qorg(0, i))
        Differ(j) = Differ(j) + D(j, i)
    Next
Next
If IterateGi = 1 Then
    For j = 1 To steps
        If Differ(j) < minls Then
            theindex = j
            minls = Differ(j)
        End If
    Next
    txtGi.Refresh:    txtGi.Text = Gi1(theindex) / 1000000000
    tolerance
    If matchGi = 1 Then
        Exit Sub
    Else
        Automatic
    End If
Else
    For j = 1 To steps
        If Differ(j) < minls Then
            theindex1 = j
            minls = Differ(j)
        End If
    Next
    txtA.Refresh:    txtA.Text = A(theindex1)
    Bx = (DeltaMPi / QiQ ^ 2) - (A(theindex1) / QiQ)
    txtB.Refresh:    txtB.Text = Bx
    txtfndi.Refresh: txtfndi.Text = Fndi(theindex1)
    tolerance
    If matchA = 1 Then
        Exit Sub
    Else
        If Check1 = 0 Then
            Automatic
        End If
    End If
End If
End Sub

```

```

Public Sub tolerance()
If IterateGi = 1 Then
    ZGi = ZGi + 1
    countG = countG + 1
    If Abs(minls - MinlsGi) < 1 Then
        matchGi = 1
        IterateGi = 0
        MinlsGi = 0
    Else

```

```

        matchGi = 0
        MinlsGi = minls
    End If
    txtiterateGi.Refresh: txtiterateGi.Text = countG
Else
    ZA = ZA + 1
    countA = countA + 1
    If Abs(minls - MinlsA) < 1 Then
        matchA = 1
        MinlsA = 0
        Exit Sub
    Else
        matchA = 0
        MinlsA = minls
    End If
    txtiterateA.Refresh: txtiterateA.Text = countA
End If
If matchGi = 1 Then
    matchGi = 0
    Automatic
End If
End Sub

Public Sub Automatic()
If IterateGi = 1 Then
    Gi1(0) = Gi1(1) * 0.9: Gi1(11) = Gi1(10) * 1.1: MinGi = Gi1(theindex - 1)
    MaxGi = Gi1(theindex + 1): steps1 = (MaxGi - MinGi) / steps
    For j = 1 To steps
        Gi1(j) = MinGi + (j - 1) * steps1
    Next
    WorkOnGiA
Else
    If (ZA = 0) Then
        MinA = DeltaMPi / (QiQ * 10): MaxA = DeltaMPi / (QiQ * 1.1)
    Else
        A(0) = A(1) * 0.1: MinA = A(theindex1 - 1)
        MaxA = A(theindex1 + 1): steps1 = (MaxA - MinA) / steps
    End If
    For j = 1 To steps
        A(j) = MinA + (j - 1) * steps1
    Next
    WorkOnGiA
End If
End Sub

Public Sub Interpolation2()
Dim Tx2, Tx1, diff, diff1, ratio, dif, k2 As Single
Dim Q1(20, 1001), Q2(20, 1001), T1(20, 1001), T2(20, 1001), Va(20, 1001) As Single
'Find the closest values to the original time from each generated Curve
For j = 1 To steps
    k2 = 1
    Do While (k2 < Ndata + 1)
        Tx2 = 1000000
        For i = 1 To XSpan
            Tx1 = Abs(Time(0, k2) - Tp(j, i))
            If Tx1 < Tx2 Then
                Tp1(j, k2) = Tp(j, i)
                Qp1(j, k2) = Qp(j, i)
                Va(j, k2) = i
                Tx2 = Tx1
            Else
                Exit For
            End If
        Next
        k2 = k2 + 1
    Loop
Next

```

```

' Check1 if the closest values are greater or smaller than the original time
For j = 1 To steps
    For i = 1 To Ndata
        If ((Time(0, i) - Tp1(j, i)) < 0) Then
            Qp(j, 0) = QiQ
            T2(j, i) = Tp(j, Va(j, i)): T1(j, i) = Tp(j, Va(j, i) - 1):
            Q2(j, i) = Qp(j, Va(j, i)): Q1(j, i) = Qp(j, Va(j, i) - 1)
        Else
            Qp(j, 0) = QiQ
            T1(j, i) = Tp(j, Va(j, i)): T2(j, i) = Tp(j, Va(j, i) + 1):
            Q1(j, i) = Qp(j, Va(j, i)): Q2(j, i) = Qp(j, Va(j, i) + 1)
        End If
    Next
Next

' Do the interpolation
For j = 1 To steps
    For i = 1 To Ndata
        diff = T2(j, i) - T1(j, i): diff1 = Time(0, i) - T1(j, i)
        ratio = diff1 / diff
        dif = Q1(j, i) - Q2(j, i)
        Qn(j, i) = (Q1(j, i) - (ratio * dif))
    Next
Next
End Sub

Public Sub WorkOnGiA()
    Dim count As Integer
    For j = 1 To 10
        Volume(j) = Gi1(j) * Bgi
        XSpan = 0: count = 0: CumDays = 0: CumGpScf = 0
        'To find the number of steps
        For i = 0 To 5000
            count = count + 1
            Pnow = MaxP - i * DeltaPressure
        Next
    Next
End Sub

' Reservoir Properties (R)(starting from the Maximum pressure which is equal to initial reservoir pressure)
' read the pressure and z from the file
R = Pnow / DeltaPressure
Pnow = PRead(R): Znow = ZRead(R): MPR = MPRead(R): ViscR = VRead(R)

'Calculation of flow Rate (Q) from Deliverability Equation
ViscAvg = (ViscR + ViscWF) / 2
DeltaMP = MPR - MPWF

' lower limit of the pressure
If count = 1 Then
    DeltaMPi = DeltaMP
End If

' initial guess of A
If IterateGi = 1 Then
    If Check1 = 0 Then
        A(j) = DeltaMPi / (QiQ * Fndii(k)) ' first arbitary geuss of fndi = 1.77
    Else
        A(j) = DeltaMPi / (QiQ * Fndi(theindex1))
    End If
End If

' use Material balance equation to calculate the Gp
If count = 1 Then
    b = (DeltaMPi / QiQ ^ 2) - (A(j) / QiQ)
    Fndi(j) = 1 + b / A(j) * QiQ
End If
MQ = Sqr(A(j) ^ 2 + 4# * b * DeltaMP)
Q = (-A(j) + MQ) / (2# * b)
Qnow = Q
If (Qi = 0) Then GoTo 5000
Qavg = (Qnow + Qi) / 2

```



```

Material Balance Equation
Bgi = 0.02829 * Zi * (Temp + 460#) / Pi
Bgf = 0.02829 * Znow * (Temp + 460#) / Pnow
If IterateGi = 0 Then
    Gi = Volume(theindex) / Bgi
Else
    Gi = Volume(j) / Bgi
End If
GpScf = (Gi * (Bgf - Bgi)) / Bgf: GpMcf = GpScf / 1000#
CumGpScf = CumGpScf + GpScf
TDays = GpMcf / Qavg: CumDays = CumDays + TDays
Qp(j, i) = Qavg: Gp(j, i) = CumGpScf: Tp(j, i) = CumDays
Gp1Mcf(j, i) = CumGpScf / 1000 ' in Mcf

5000 Qi = Qnow
      Zi = Znow
      Pi = Pnow

If ((Pnow - DeltaPressure) < Low) Then
    Exit For
End If
XSpan = XSpan + 1
Next
Bgi = 0.02829 * ZRead(R1) * (Temp + 460#) / MaxP
Qi = 0
Next
Interpolation2
leastsquare
End Sub

Public Sub FndiRange()
Fndii(0) = 1
For k = 1 To 10
    Fndii(k) = Fndii(k - 1) + 0.25
Next
Fndii(0) = 1.1: k = 0
End Sub

Public Sub GetResult()
min(k) = minls: Af(k) = A(theindex1): Fndif(k) = Fndi(theindex1)
Gif(k) = Gi1(theindex) / 1000000000: Bf(k) = b
For i = 1 To XSpan
    Tf(k, i) = Tp(theindex1, i): Qf(k, i) = Qp(theindex1, i): Gpf(k, i) = Gp1Mcf(theindex1, i)
Next
End Sub

Public Sub Clean()
Dim C As Integer
With grdDecline
For C = 1 To Ndata
    .Col = 1: .row = C: .Text = ""
    .Col = 2: .row = C: .Text = ""
    .Col = 3: .row = C: .Text = ""
Next
End With
End Sub

```