INVISIBLE GOLD INC.

PNGE 295 SENIOR DESIGN

SENIOR ENGINEER
Michael Hupp

DESIGN ENGINEERS
Alan Camponeschi
Humberto Ramirez

April 30, 1999

EXECUTIVE SUMMARY

Two wells were chosen in Clay County, WV. One of the wells is located in the Elkhurst quadrangle (**figure 1.1.3**), and the other is located in the Strange Creek quadrangle (**figure 1.1.4**). The Elkhurst well contained one target formation (the Weir), and the Strange Creek well contained two possible producing formations (Blue Monday and Big Injun). All of the formations are Mississipian Aged rocks consisting of Sandstone matrices. The Elkhurst well has three coal seams that could prove to be harmful to the life of this well.

After confirming that both formations showed potential to be good producers based on geological data, several logs were run to determine the porosity and the gas saturation of the Strange Creek and Elkhurst wells. The logs used are as follows: Neutron Porosity, Density Porosity, Density, Resistivity, and Gamma Ray logs. A Nuclear Survey accompanied the logs for the Strange Creek well. Nuclear Survey is a special tool used by a company to simplify the log interpretation process. The porosity value for the Weir formation in the Elkhurst well was determined to be approximately 12.0%. The Initial Gas in Place and Estimated Ultimate Recovery for the Elkhurst were calculated to be 1623 Mcf/acre and 1379.6 Mcf/acre respectively. The porosity values determined in the Strange Creek well for the Big Injun (Big Lime) was 5.0%, the Blue Monday was 10.0%, and the prospect formation (the Maxton Sand) was approximately 7.0%. The total Initial Gas in Place and the Estimated Ultimate Recovery for the two producing formations was calculated to be 6186.6 Mcf/acre and 5258.6 Mcf/acre respectively. When the prospect zone is added to the cumulative totals in the Strange Creek well, the total Initial Gas in Place becomes 6486.4 Mcf/acre and the Estimated Ultimate Recovery becomes 5513.4 Mcf/acre.

Following the determination of Initial Gas in Place (IGIP) and Estimated Ultimate Recovery (EUR), the permeability and damage was calculated. Build-up data obtained for the formations (Weir, Blue Monday, and Big Injun) was gathered from example problems in the Petroleum and Natural Gas Engineering 271 class at West Virginia University. This build-up data was then manipulated by multiplying the series of data by a constant factor. This allowed for more representative data associated with the initial reservoir pressures of the three formations. The reservoir data used in the calculation process was data obtained through the previous steps of the project (ie. porosity from logs, reservoir temperature from logs, etc…).

The final calculated values for the permeability and skin factor in the formations located in the two wells was as follows:

Well	Formation	k (mD)	Skin	Wellbore Storage
Elkhurst	Weir	0.4	10.28	3,914
Strange Creek	Big Injun	1.27	13.48	1,777
	Blue Monday	1.59	-5.68	298,191

A production schedule was then determined through the optimization of the flow rate for each producing zone. The properties of each reservoir were used in evaluating a constant flow rate that could be sustained until the end of the contract period. The project was desired to have a total life of seven years based on the contract with the Shahab

Mohaghegh, Inc. This feat was accomplished through the use of the deliverability equation in conjunction with the average compressibility factor, average temperature approach for calculating wellhead pressure.

The results for the optimum flow rate are as follows:

Well	Formation	Flow Rate (Mcfd)	Time (yr)	Time (days)
Strange Creek	Blue Monday	0.79	7.01	2558.65
	Big Injun	1.41	7.01	2558.65
Elkhurst	Weir	0.59	7.01	2558.65

Next, a plot of viscosity versus pressure was developed, and a polynomial equation was fitted to the plot. The three polynomials are as follows for the three formations:

Big Injun: $y = (1E\text{-}09)x^2 + (7E\text{-}07)x + 0.0108$
Blue Monday: $y = (1E\text{-}09)x^2 + (8E\text{-}07)x + 0.0108$
Weir: $y = (1E\text{-}09)x^2 + (6E\text{-}07)x + 0.0108$

As can be seen from the polynomial equations, all three of the plots seem to have an almost identical relationship.

Finally, both projects were evaluated through the use of Monte Carlo simulation to deem which project would be most profitable. Based on the scenarios for the uncertainties of the reservoir parameters, the following distinction was made. The Discount Cash Flow Rate of Return for the Strange Creek well was higher, thus the Strange Creek well will be the most profitable venture. Therefore, Invisible Gold, Inc. will be pushing ahead for the development of the Strange Creek area.

TABLE OF CONTENTS

INTRODUCTION

This report deals with the semester long task of developing a reservoir from start to finish. The many stages associated with the development of a reservoir are thoroughly described in this report. The basis for the project was to deem which of the two wells would be the most profitable investment.

The first stage for reservoir development is, obviously, finding a reservoir. The reservoirs used in this project were associated with wells already present in West Virginia. The corresponding geology of the formations found in these wells was then researched and evaluated.

At this point, the next step was designing the casing string and cement job to be used in each of the wells. This is a crucial part of the reservoir development. If the design is not a solid design, problems could occur which could drastically affect the life of the well in question. The better the design, the lower the workover costs for the well. The lower the workover costs, the longer the well will be economical to produce.

Upon researching the formation rock and designing the casing string, logs were evaluated from the wells in question. The logs were essential in determining the porosity, thickness, and gas saturation values for each of the formations. From these interpreted values, an estimate for the initial gas in place for each of the reservoirs was calculated. This estimate was done on a scf per acre basis, assuming a drainage area of 52 acres for each of the wells.

The next step in the process was gathering build-up data in order to evaluate the permeability and damage (skin) associated with each of the formations. This involved using all of the data collected in the previous steps to the development of each well and corresponding reservoir(s). These values for permeability and skin allowed for the determination of possible future stimulation treatments, if applicable.

An optimum flow rate was then determined based on the damage and permeability, in conjunction with the other determined reservoir parameters, in the formation. The flow rate was to be a constant rate to be produced over the pre-determined contract period (seven years).

Finally, the DCFROR for each project was determined to find the most profitable investment. The project with the highest DCFROR would be the project of choice, or the best investment.

CHAPTER 1: THE GEOLOGY

The Strange Creek reservoir is located in Clay County, WV and has an interesting geological outlook. The formation is made up of Early to Middle Mississipian Age rock. The formation is part of a bigger group of formations called the Greenbrier Group. The name of the target formation is the Price Formation and the rock matrix is sandstone. The genetic make of the reservoir is heterogeneity caused by diagenetic deposition. Diagenetic deposition is the build up of sediments that look like layers. The reservoir is the result of a combination trap. A combination trap is the result of a stratigraphic and structural trap. For example there can be a fault (structural trap) and depositional trap (stratigraphic trap) leading up to the fault. A depositional trap is caused by porosity and permeability values decreasing as the move up in the formation until the can no longer continue up and started to collect and pool together in the formation. The primary producing reservoir is the Big Injun (Big Lime). There is a possible secondary-producing reservoir that was discovered while drilling for the Big Injun. This reservoir is termed the Blue Monday. The drive mechanism of the well is gas expansion.

Strange Creek				
Reservoir	Depth (ft)	Total Area (acres)	Pay zone (ft)	Average Porosity
Big Injun	1831	14,710	26	15%
Blue Monday	1665	-	24	13%

FIGURE 1.1.1

The Elkhurst well is also located in Clay County, WV. The reservoir is composed of Early and Lower Mississipian Age Rock. The formation is part of a larger group termed the Pocono Group. The formation that was penetrated is known as the Price formation. The rock matrix of this area is sandstone. The reservoir holds the natural gas in place by using a combination trap (look to previous for definition). The make up of the reservoir is that of heterogeneity caused by deposition. The drive mechanism for this well is gas expansion. There were three coal seams that were discovered while drilling the well.

Elkhurst				
Reservoir	Depth (ft)	Total Area (acres)	Pay zone (ft)	Average Porosity
Weir	1888	4,073	31	9.30%

FIGURE 1.1.2

Elkhurst
Reservoir: Weir

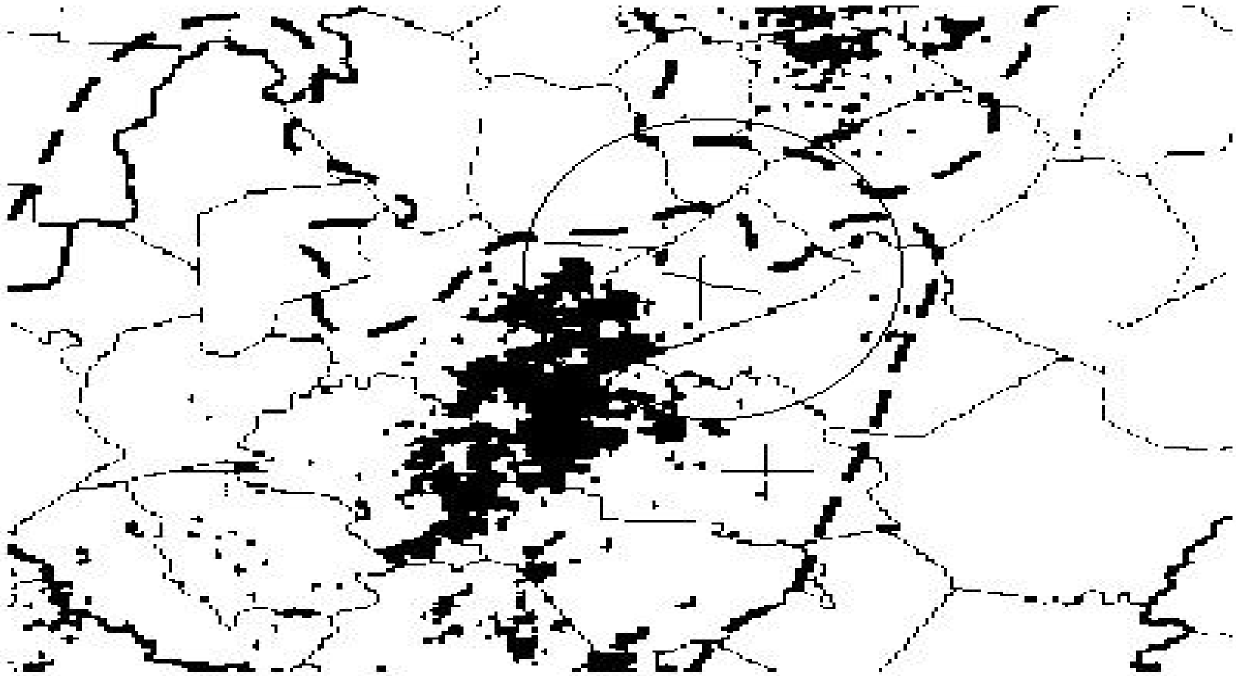


Figure 1.1.3

Strange Creek
Reservoir: Big Injun

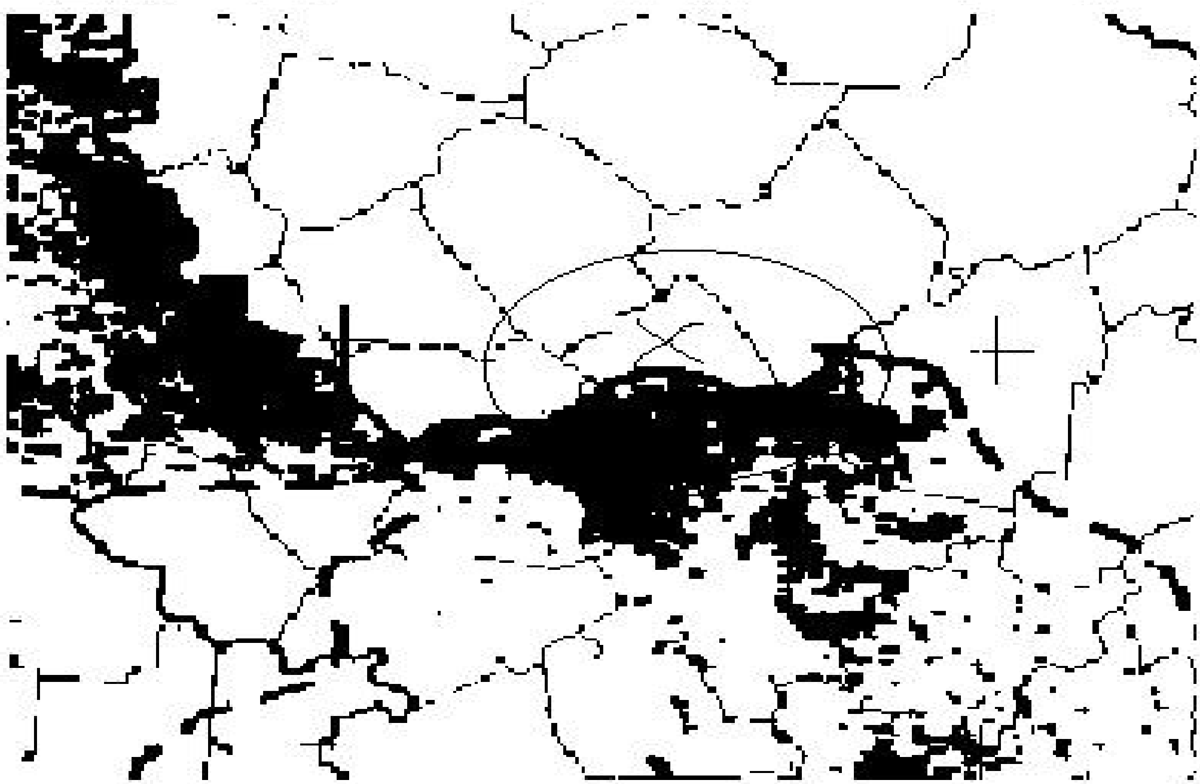


Figure 1.1.4

CHAPTER 2: THE COMPLETION

I. Background and Theory to Casing Design:

Casing serves several important functions in drilling and completing a well. It prevents collapse of the borehole during drilling and hydraulically isolates the wellbore fluids from the subsurface formation and formation fluids. It minimizes damage of both the subsurface environments by the drilling process and the well by a hostile subsurface environment. It provides a high – strength flow conduit for the drilling fluid to the surface and, with the blowout preventers (BOP) permits the safe control of formation pressure. Selective perforation of properly cemented casing also permits isolated communication with a given formation of interest.

Casing is defined as tubular pipe with a range of OD's of 4.5" to 20". Among the properties included in the API standards for both pipe and couplings are strength, physical dimensions, and quality-control test procedure. In addition to these standards, API provides bulletins on the recommended minimum-performance properties must be used in the design of casing string to minimize the possibility of casing failure.

API has adopted a casing *grade* designation to define the strength characteristic of the pipe. The grade code consists of a letter followed by a number. The letter designation in the API grade was selected arbitrarily to provide a unique designation for each grade of casing adopted in the standards. The number designates the minimum yield strength of the steel in thousands of psia. API defines the yield strength of the steel as the tensile stress required to produce a total elongation per unit length of 0.005 on standard test specimen.

Casing dimensions can be specified by casing (OD) and nominal wall thickness. However, it is conventional to specified casing dimensions by size and weight per foot. In discussing casing weight, one should differentiate between nominal weight, plain –end weight, and average weight for threads and couplings. The nominal weight per foot is no a true weight per foot but it is useful for identification purposes as an approximate average weight per foot. The pain end weight per foot is weight per foot is the weight per foot of the pipe body, excluding the threaded portion and coupling weight. The average weight per foot is the total weight of an average joint of the treaded pipe, with a coupling attached power-tight at one end, divided by the total length of the average joint.

API provides specifications for the following for types of casing connectors:

- short round threads and couplings (CSG)
- long round threads and couplings (LCSG)
- buttress threads and couplings (BCSG)
- extreme line threads (XCSG)

The CSG and LCSG Connectors have the same basic thread design. Threads have a rounded shape and are spaced to give eight threads per inch. The threads are cut with a taper of ¾ in./ft on diameter for all pipe sizes. A longer thread run out and coupling of the LCSG provide a greater strength when needed. The BCSG joint efficiency of this connector is 100% in most cases. The design thread is very similar as the round thread. However longer coupling and thread run-out are used and the shape thread is squarer, so the unzipping tendency is greatly reduced. The XCSG differs from the other connectors in that it is an integral joint. On an integral joint the pipe wall must be thicker near the ends of the casing to provide the necessary metal to machine a stronger connections.

API CASING PERFORMANCE PROPERTIES

The most important performance properties of casing include its rated values for axial tension, burst pressure, and collapse pressure.

TENSION

Axial tension loading results primarily from the weight of the casing string suspended below the joint of interest. Body yield strength is the tensional force required to exceed the elastic limit of the pipe body. Similarly, joint strength is the minimum tensional force required causing joint failure. (Figure 2.1.1)

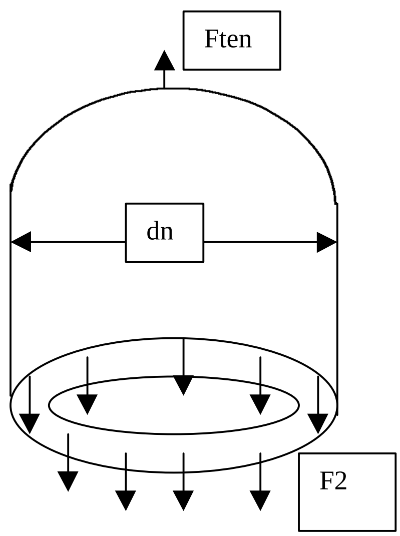


Figure 2.1.1

For Ften tending to pull apart the pipe is resisted by the strength of the pipe wall which exert a counter force F2

COLLAPSE PRESSURE:

Collapse pressure is the minimum external pressure that will cause the casings walls to collapse in the absence of external pressure and axial loading. (Figure2.2.2)

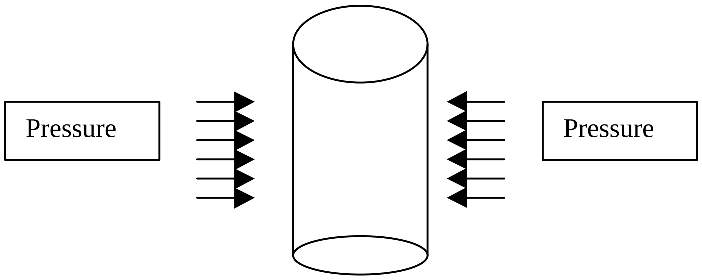


Figure 2.1.2

BURST PRESSURE:

The burst pressure is the minimum internal pressure that would cause the casing to rupture in the absence on external pressure and axial loading. (Figure 2.1.3)

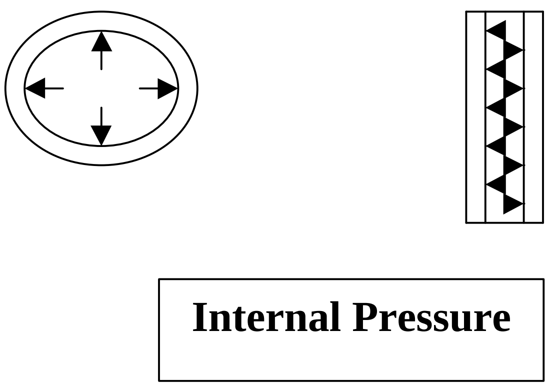


Figure 2.1.3

II. Casing Design

The casing program is accomplished from 3 steps. In the first step, the casing sizes and corresponding bit sizes should be determined. In the second step, the setting depths of individual casing string ought to be evaluated, and the last step which is the determination on the considerations of loads on the string. Before starting the casing program design, The designer ought to know the following basic information:

- The purpose of the well (exploratory or development drilling);
- Geological cross-sections that should consists of type of formation, expected hole problem, pore and formation fracture pressure, number and depth of water , oil, and gas horizons;
- Available rock bits and casing sizes;
- Load capacity of a derrick and mast if the type of rig has already been selected.

Before staring the design, it must be assumed that the production casing size and depth of the well have been established by the petroleum engineer in cooperation with a geologist, so that the hole size (rock bit diameter) for the casing must be selected. Considering the diameter of the hole, a sufficient clearance beyond the coupling outside diameter must be provided to allow for mud cake and also for a good cementing job. Fields experience shows that the casing clearance should range from a about 1.0 in to 3.5 in. Once the size of production string has been selected, the smallest casing through which a given bit will pass is next determined. The bit diameter should be a little less (0.05 in) than drift casing diameter. After choosing the casing with appropriate drift diameter, the outside coupling diameter of this casing may be found. Next, the appropriate size of the bit should be determined and the procedure repeated.

Step1 (well 1 – ELKHURST CASING CLEARANCE)

Bit size, in	OD casing,in	Weight, lb/ft	Grade	Wall thickness, in	Drift, in	OD coupling, in	Range from 1 to 3, in
12-1/4	8-5/8	28.00	H-40	0.304	7.892	9.625	2.625
6-3/4	4-1/2	9.5	H-40	0.205	3.965	5.0	1.75

Step 2 (DEPTH SELECTION)

The selection of the number of casing strings and their respective setting depths generally is based on a consideration of the pore pressure gradient and fracture gradient of the formation to be penetrated. The pore pressure and the fracture pressure gradient are expressed in equivalent density, and are plotted Vs depth. Here is the graph that estimates how many casing are needed and its respective depths. A graphical representation is shown on Fig.2.2.1

WELL 1 (ELKHURST)

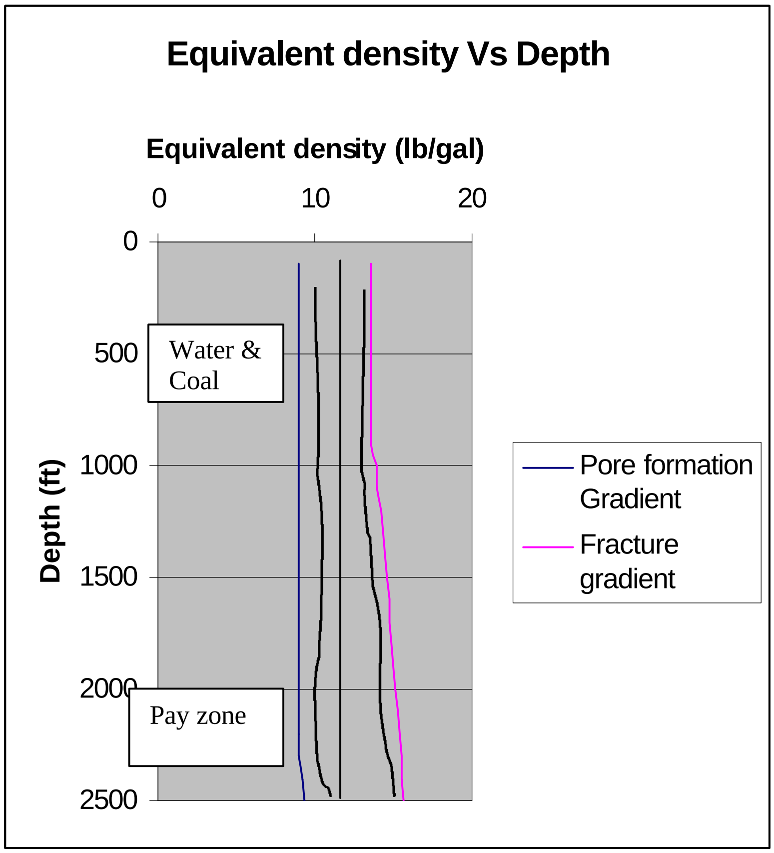


Fig.2.2.1 (Depth Selection)

As you can see from this graph that only one casing could be used, but for protection of fresh water aquifers and presence of coal, a surface casing was needed at 830'.

Step 3 (LOAD CONSIDERATIONS- ELKHURST)

TENSION:

For a casing: 8-5/8 H-40 28.00 lb/ft (Round thread joint)
and
For a casing: 4-1/2 H-40 9.5 lb/ft (Round thread joint)

OD,in	Grade	Weight, lb/ft	Pipe strength, lbf	Joint strenght, lbf	W * Depth lbs	Pipe strength /2 lbs	Joint strenght/2 lbs	Safety Factor 1.6
8-5/8	H-40	28.00	318,000	233,000	23,240	159,000	116,500	6.8-5.0
4-1/2	H-40	9.50	111,000	77,000	23,750	55,500	38,500	2.3-1.6

If safety factor greater than 1.6 then the string is OK.

BURST PRESSURE:

Pf = 0.052*(13.1)*830' = 610 psia [Fracture pressure]

The surface casing pressure for the design loading conditions is:

610 – 0.05*(830') = 570 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(830') = 380 psia

The pressure differential that tends to burst the casing is 570 psia at the surface, and 230 psia (610 – 380) at the casing seat. Multiplying this pressure by a safety factor of 1.1 yields a burst design load of 630 psia at the surface, and 255 psia at the casing seat. A graphical representation of the burst-design is shown in **Fig.2.2.2.**

OD,in	Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
8-5/8	**H-40**	**28.00**	**2,470**	**255**

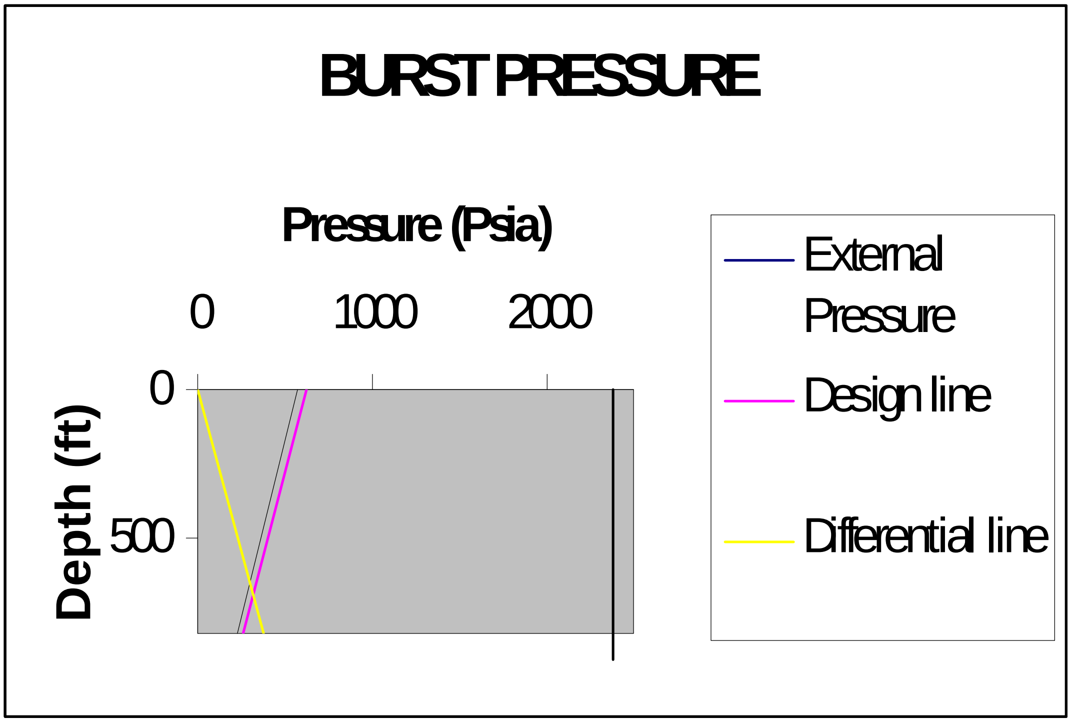


Fig.2.2.2 (Casing set @ 830')

COLLAPSE PRESSURE

The design for collapse is based on the density of mud in the hole when the casing is being run. According to the graph in step 2 one can select the mud density, and the external pressure at 830' is (assume water in the hole 8.33 lb/gal + safety factor of 0.6 lb/gal)

0.052(8.9)(830') = 385 psia

The internal pressure for the collapse – design load is controlled by the maximum loss in fluid level that could occur if a severe loss of circulation problem is encountered. The way this is designed is by assuming the worst scenario where a loss of circulation zone is unexpectedly encountered near the depth of the next casing seat. The differential pressure that tends to collapse the casing is zero at the surface.

Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 420 at 830'. A graphical representation of the collapse load is shown on **Fig.2.2.3.**

OD,in	Grade	Weight, lb/ft	Collapse resistance, psia	External pressure + SF, psia
8-5/8	**H-40**	**28.00**	**1,610**	**420**

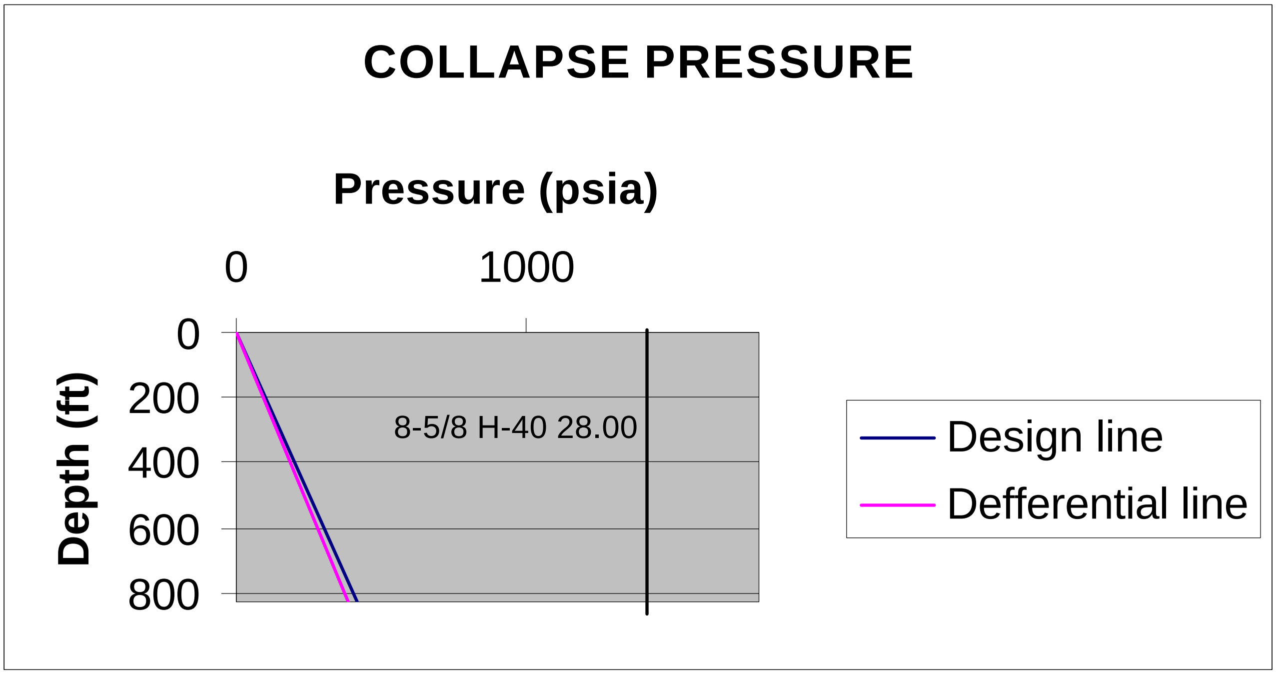


Fig.2.2.3 (Casing set @ 830')

PRODUCTION CASING: 4-1/2 H-40 9.5 lb/ft

For step 1 and 2 they were previously done, and for step 3 (Tension) was also done in the first part.

BURST PRESSURE

Pf = 0.052*(14)*2295' = 1670 psia [Fracture pressure]

The surface casing pressure for the design loading conditions is:

1670' – 0.05*(2295') = 1555 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(2295') = 1070 psia

The pressure differential that tends to burst the casing is 1555 psia at the surface, and 600 psia (1670 –1070) at the casing seat. Multiplying this pressure by a safety factor of 1.1 yields a burst design load of 1710 psia at the surface and 660 psia at the casing seat. A graphical representation of the burst load is shown in **Fig.2.2.4.**

OD,in	Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**3,190**	**1070**

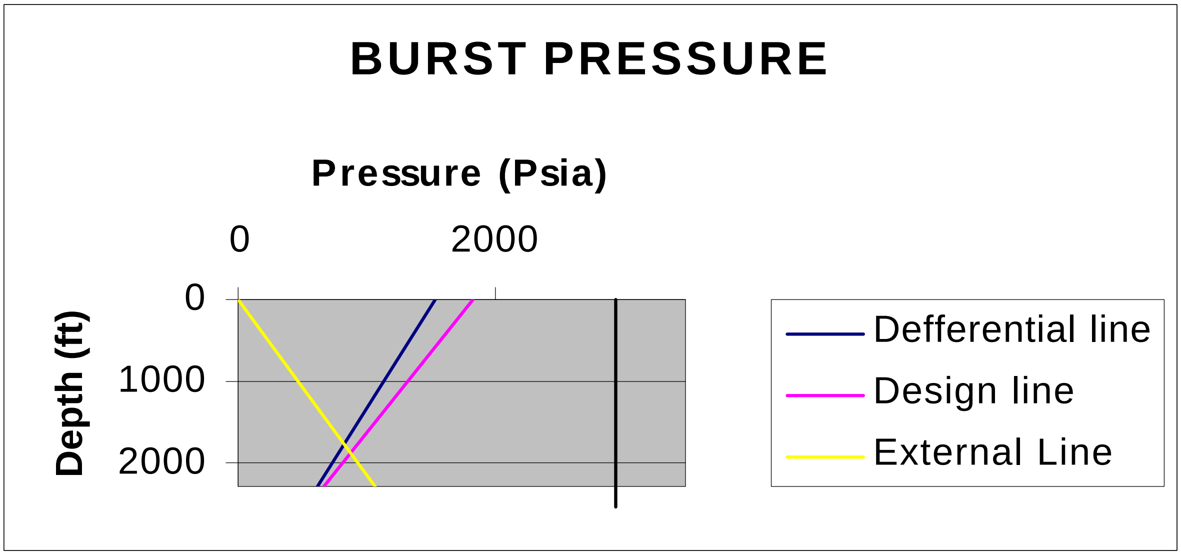


Fig.2.2.4 (casing set at 2295'*)*

COLLAPSE PRESSURE:

The design for collapse is based on the density of mud in the hole when the casing is being run. According to the graph in step 2, one can select the mud density and the external pressure at 2295' is (assume water in the hole 8.33 lb/gal + safety factor of 0.6 lb/gal)

0.052(8.9)(2295') = 1060 psia

The internal pressure for the collapse – design load is controlled by the maximum loss in fluid level that could occur if a severe lost of circulation problem is encountered. The way this is designed by assuming the worst Scenario where a lost of circulation zone unexpectedly is encountered near the depth of the next casing seat. The differential pressure that tends to collapse the casing is zero at the surface
Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 1160 at 2295'. A graphical representation of the collapse load is shown on **Fig.2.2.5.**

OD,in	Grade	Weight, lb/ft	Collapse resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**2,760**	**1160**

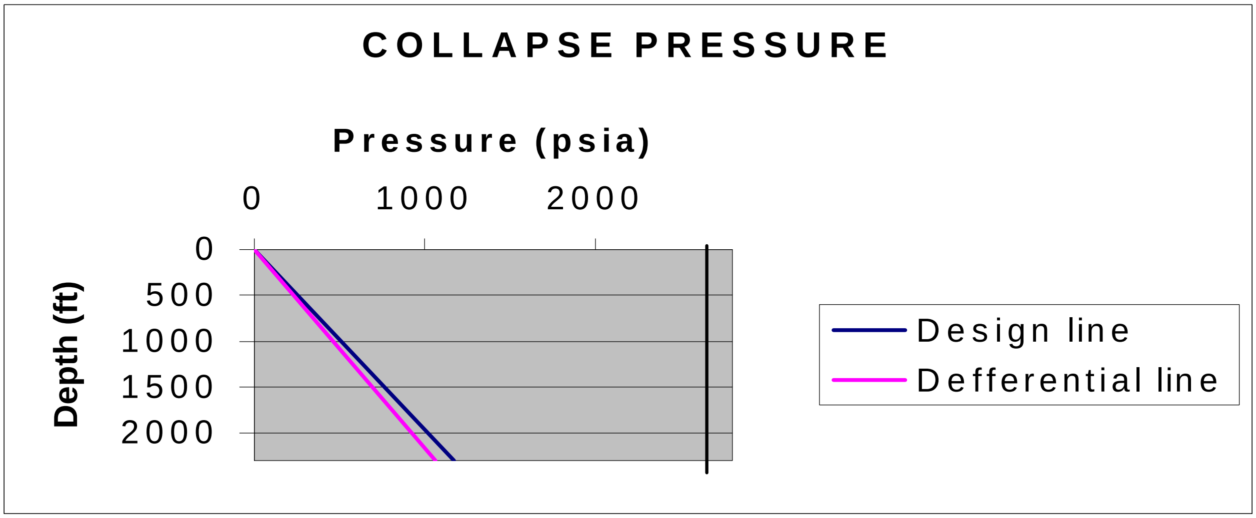


Fig. 2.2.5

Step1 (well 2 – STRANGE CREEK CASING CLEARANCE)

Bit size, in	OD casing,in	Weight, lb/ft	Grade	Wall thicknes s, in	Drift, in	OD coupling , in	Range from 1 to 3, in
8-3/4	7	17.00	H-40	0.231	6.413	7.656	1.094
6-3/4	4-1/2	9.5	H-40	0.205	3.965	5.0	1.75

Step 2 (DEPTH SELECTION)

The selection of the number of casing strings and their respective setting depths generally is based on a consideration of the pore pressure gradient and fracture gradient of the formation to be penetrated. The pore pressure and the fracture pressure gradient are expressed in equivalent density, and are plotted Vs depth. Here is the graph that estimates how many casing are needed and its respective depths. A graphical representation is shown on **Fig.2.2.6.** (*)

WELL 2 (STRANGE CREEK)

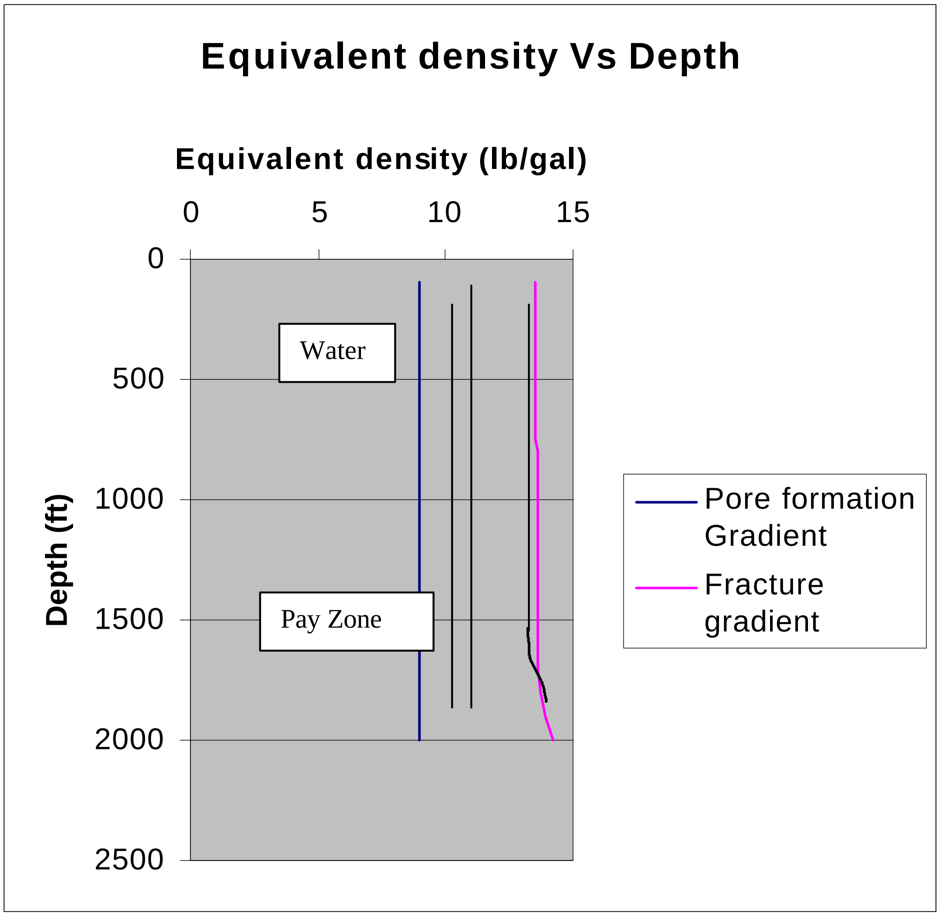


Fig.2.2.6 (Depth selection)
(*) As you can see from the graph that says that only one casing can be run, but again for protection of fresh water aquifers a surface casing has to be run @ 310'.

Step 3 (LOAD CONSIDERATIONS - ELKHURST)

TENSION:

For a casing: 7" H-40 17.00 lb/ft (Round thread joint)
and
For a casing: 4-1/2" H-40 9.5 lb/ft (Round thread joint)

OD, in	Grade	Weight, lb/ft	Pipe strength, lbf	Joint strenght, lbf	W * Depth lbs	Pipe strength /2 lbs	Joint strenght/2 lbs	Safety Factor 1.6
7	H-40	17.00	196,000	122,000	5,270	98,000	61,000	18.5-11.5
4-1/2	H-40	9.50	111,000	77,000	23,750	55,500	38,500	2.3-1.6

If safety factor greater than 1.6 then the string is OK.

BURST PRESSURE:

Pf = 0.052*(13.1)*310' = 210 psia [Fracture pressure]

The surface casing pressure for the design loading conditions is:

210 – 0.05*(310') = 195 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(310') = 145 psia

The pressure differential that tends to burst the casing is 195 psia at the surface, and 65 psia (210 –145) at the casing seat. Multiplying this pressure by a safety factor of 1.1 yields a burst design load of 215 psia at the surface and 70 psia at the casing seat. A graphical representation of the burst-design is shown **Fig.2.2.7.**

OD,in	Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
7	**H-40**	**17.00**	**2,310**	**215**

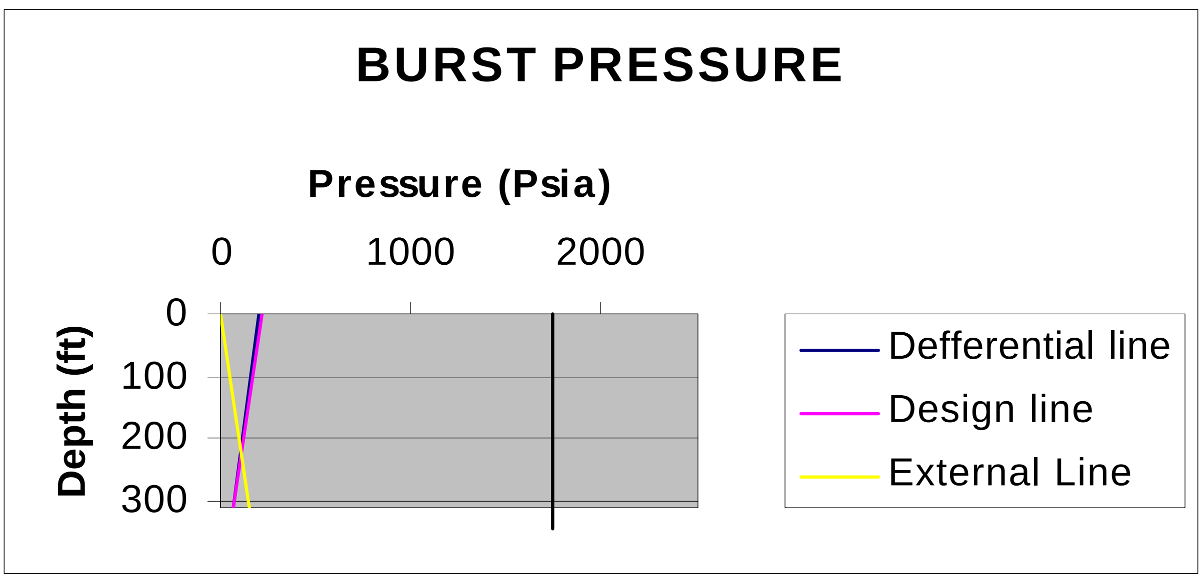


Fig.2.2.7 (Casing set @ 310')

COLLAPSE PRESSURE:

The design for collapse is based on the density of mud in the hole when the casing is being run, according to the graph in step 2, you can select the mud density and the external pressure at 310' is (assume water in the hole 8.33 lb/gal + safety factor of 0.6 lb/gal)

0.052(8.9)(310') = 145 psia

The internal pressure for the collapse –design load is controlled by the maximum loss in fluid level that could occur if a severe lost of circulation problem is encountered. The way this is designed by assuming the worst Scenario where a lost of circulation zone unexpectedly is encountered near the depth of the next casing seat.

The differential pressure that tends to collapse the casing is zero at the surface Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 160 at 310'. A graphical representation of the collapse load is shown on Fig.2.2.7

OD,in	Grade	Weight, lb/ft	Collapse resistance, psia	External pressure + SF, psia
7	**H-40**	**17.00**	**1,420**	**160**

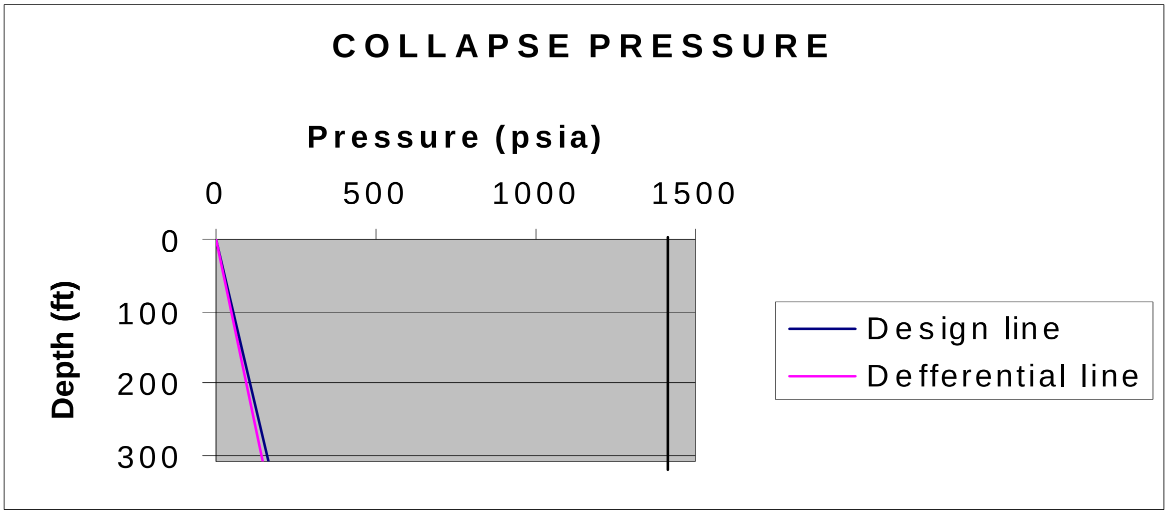


Fig.2.2.7 (Casing set @ 310')

7" H-40 17.00 lb/ft

***PRODUCTION CASING*: 4-1/2 H-40 9.5 lb/ft**

For step 1 and 2 they were previously done, and for step 3 (Tension) was also done in the first part.

BURST PRESSURE:

Pf = 0.052*(13.2)*1965' = 1350 psia [Fracture pressure]

The surface casing pressure for the design loading conditions is:

1350' – 0.05*(1965') = 1250 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(1965') = 900 psia

The pressure differential that tends to burst the casing is 1250 psia at the surface, and 450 psia (1350 – 900) at the casing seat. Multiplying this pressure by a safety factor of 1.1 yields a burst design load of 1480 psia at the surface, and 495 psia at the casing seat. A graphical representation of the burst load is shown in Fig.2.2.8

OD,in	Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**3,190**	**1480**

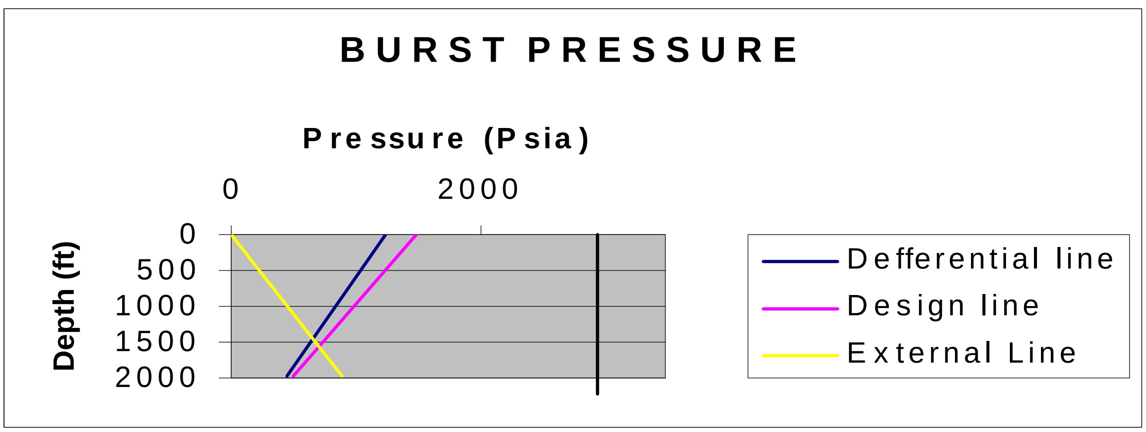


Fig.2.2.8 (Casing set @ 1965')

COLLAPSE PRESSURE:

The design for collapse is based on the density of mud in the hole when the casing is being run. According to the graph in step 2, you can select the mud density and the external pressure at 1965' is (assume water in the hole 8.33 lb/gal + safety factor of 0.6 lb/gal)

$0.052(8.9)(1965') = 910$ psia

The internal pressure for the collapse –design load is controlled by the maximum loss in fluid level that could occur if a severe lost of circulation problem is encountered. The way this is designed by assuming the worst Scenario where a lost of circulation zone unexpectedly is encountered near the depth of the next casing seat.

The differential pressure that tends to collapse the casing is zero at the surface Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 1000 at 1965'. A graphical representation of the collapse load is shown in Fig.2.2.10

OD,in	Grade	Weight, lb/ft	Collapse resistance, psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**2,720**	**1000**

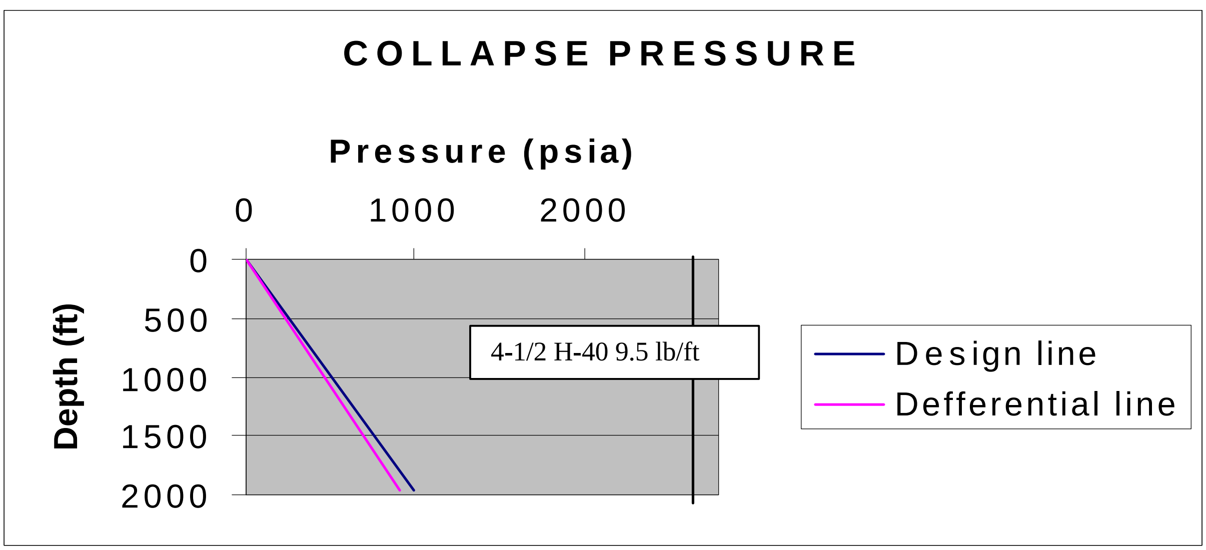


Fig.2.2.10 (Casing set @ 1965')

III. Background and Theory for Cementing Design:

The cement job is one of the most important aspects to well completion. Without a good, stable design for the cement job, problems are almost certainly going to arise. The cementing design developed for the two prospects for Invisible Gold Inc. differed from one another slightly. The prospect located in the Elkhurst quadrangle encountered three different coal seams within the first 800 feet of the well. These coal seams posed the threat of possible Sulfate problems in the future of the well. Therefore, a Class B cement was chosen for this well. This cement is somewhat more expensive than a Class A cement which was chosen to be used for the Strange Creek prospect. However, Class B cement is a good cement to use when moderate to high sulfate problems exist.

The A.P.I. designates many different grades of cement. The different classes and there properties are listed in Figure 2.3.1.

Class	Depth	Applications
A	<= 6000 ft	Special properties not required
B	<= 6000 ft	Moderate to high sulfate resistance
C	<= 6000 ft	high early strength
G	<= 8000 ft	basic cement for the given depth
H	<= 8000 ft	basic cement for the given depth; similar to class G
J	12000 to 16000 ft	Used for extremely high pressures and temperatures

Figure 2.3.1

Class A cement represents the cheapest and simplest class of cement. As the class increases, the cement tend to become more expensive, but also more durable. The strongest and most durable being classed J cement.

An additive may be mixed with the cement slurry in order to alter the properties of the cement. Different additives may increase or decrease the slurry weight. Figure 2.3.2 lists some of the additives and the properties, which the additives affect.

Additive	Properties
Bentonite	Decreases slurry weight (ppg)
NaCl	Increases slurry weight (ppg)
Kolite	Decreases slurry weight (ppg)
Diacel D	Decreases slurry weight (ppg)
Perlite	Increases slurry weight (ppg)

Figure 2.3.2

One question arose in this project as to whether an additive might be necessary in order to avoid collapse in the casing during the cement job. The answer to this question is no, and will be explained in greater detail later in this section of the report.

The height of the cement is also a very important parameter in the designing a stable cement job. Obviously, the surface casing was cemented to the surface. However, our company chose not to cement the production string to the bottom of the surface casing. The amount of water encountered in both of the prospects was fairly negligible, and no aquifers appeared to be present in the open area between the top, of the predetermined quantity, of the cement used for the production casing, and the bottom of the cement used for the surface casing. This design is more economical, and is believed to have no potential danger. This is better illustrated on the Figure 2.3.3 and Figure 2.3.4. Casing

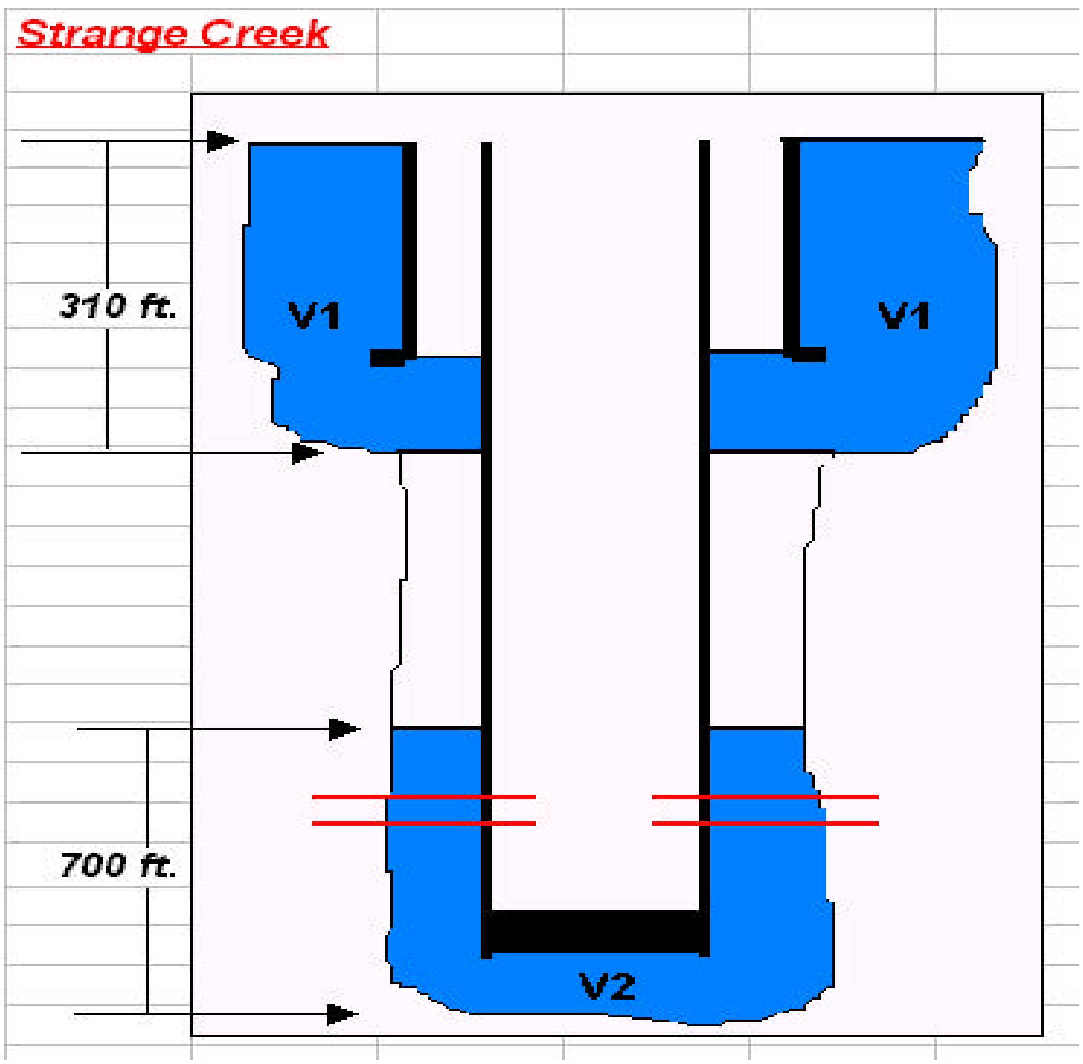


Figure 2.3.3

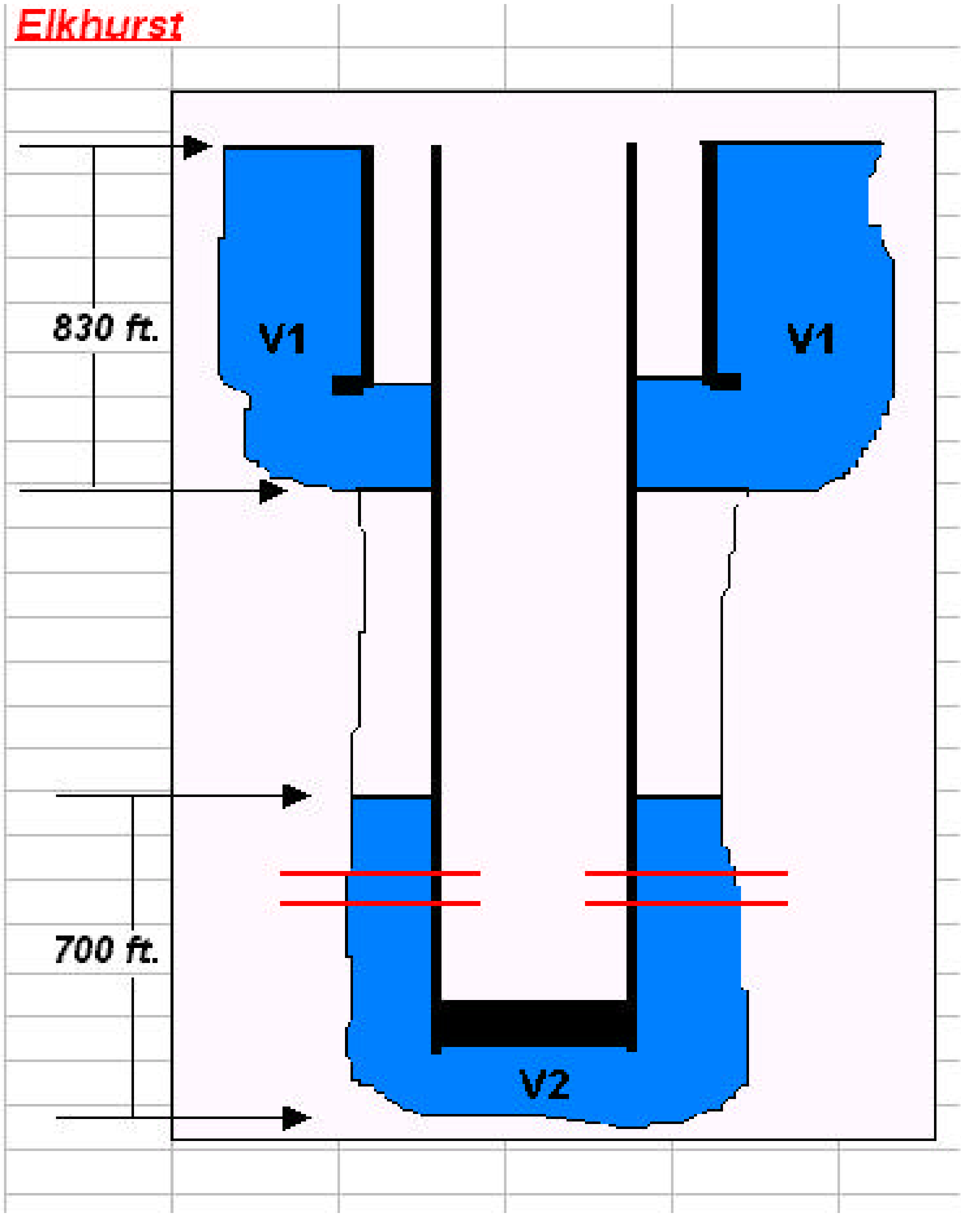
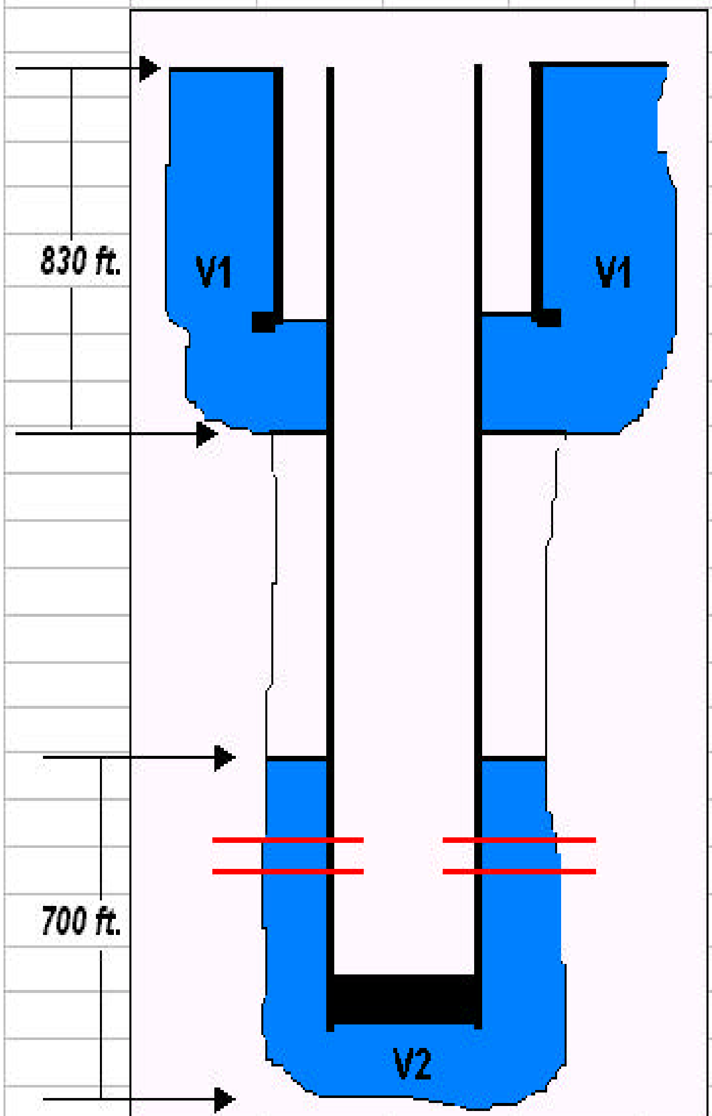


Figure 2.3.4

IV. Cementing Design:

The final results for the design of the cement job are listed in **Figure 2.4.1 and Figure 2.4.2.**

Elkhurst



V1 represents the cementing design for the surface casing.
V2 represents the cementing design for the production casi

Surface Casing (V1):

Depth (ft):	830	
Annular Area (ft^2):	0.2542	
Annular Volume (ft^3):	369	
# of Sacks of Cement:	313	
Mixing Time (hrs):	0.21	
Displacement Time (hrs):	0.064	
Setup Time (hrs):	0.25	
Safety Factor Time (hrs):	0.5	
Total Operating Time (hrs):	1.024	(min. thickening time)

Production Casing (V2):

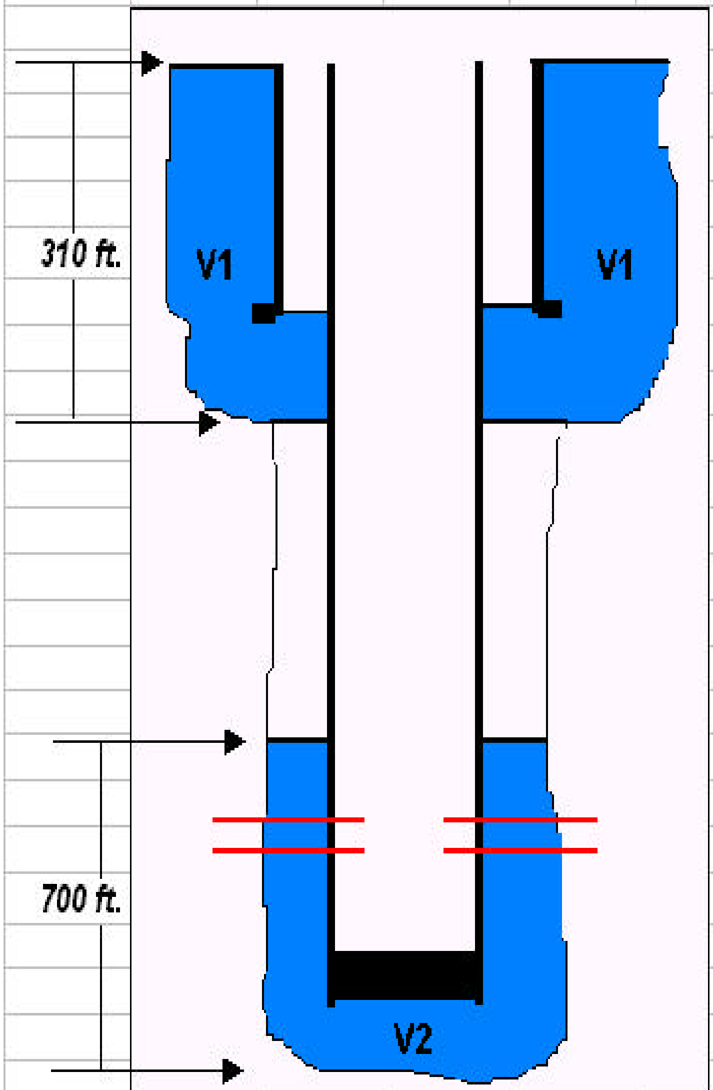
Depth (ft):	2500	
Annular Area (ft^2):	0.1381	
Annular Volume (ft^3):	169	
# of Sacks of Cement:	143	
Mixing Time (hrs):	0.095	
Displacement Time (hrs):	0.066	
Setup Time (hrs):	0.25	
Safety Factor Time (hrs):	0.5	
Total Operating Time (hrs):	0.911	(min. thickening time)

The cement is represented in blue.

comments: Class B cement was chosen due to the presence of three coal seams and water.

Figure 2.4.1

Strange Creek



V1 represents the cementing design for the surface casing.
V2 represents the cementing design for the production casing.

Surface Casing (V1):

Depth (ft):	310	
Annular Area (ft^2):	0.1385	
Annular Volume (ft^3):	75.1	
# of Sacks of Cement:	64	
Mixing Time (hrs):	0.043	
Displacement Time (hrs):	0.021	
Setup Time (hrs):	0.25	
Safety Factor Time (hrs):	0.5	
Total Operating Time (hrs):	0.814	*(min. thickening time)*

Production Casing (V2):

Depth (ft):	2000	
Annular Area (ft^2):	0.0859	
Annular Volume (ft^3):	105.2	
# of Sacks of Cement:	90	
Mixing Time (hrs):	0.06	
Displacement Time (hrs):	0.053	
Setup Time (hrs):	0.25	
Safety Factor Time (hrs):	0.5	
Total Operating Time (hrs):	0.863	*(min. thickening time)*

The cement is represented in blue.

****comments:** *Class A cement was chosen for economic purposes.*

Figure 2.4.2

The process for calculating the results displayed above is located in the Appendix.

The results of the cementing design yielded a total of 456 sacks Class B cement to be used for the prospect located in the Elkhurst quadrangle, and 154 sacks of Class A cement to be used for the prospect located in the Strange Creek quadrangle. The total time to complete the Elkhurst (time for the surface casing cement plus time for the production casing cement) was approximately 1.935 hours. The total time for the Strange Creek prospect was 1.677 hours.

A question arose as to whether or not the casing would collapse during the cement job. The answer is NO. The calculation for the pressure applied on the casing during the cementing process is as follows:

Elkhurst:

Cement weight = 15.6 lb/gal
Max. Pressure Applied on the casing = .052 * ρ * Depth
Max. Pressure Applied on the casing = .052 * 15.6 ppg * 2500ft.
Max. Pressure Applied on the casing = .2028 psi

Strange Creek:

Cement weight = 15.6 lb/gal
Max. Pressure Applied on the casing = .052 * ρ * Depth
Max. Pressure Applied on the casing = .052 * 15.6 ppg * 2000ft.
Max. Pressure Applied on the casing = .1622 psi

After comparing these numbers with the numbers obtained from the casing design for the collapse pressure, it can be observed that the casing will not collapse. In fact, the production casing could be cemented the entire length of the casing without the casing having the potential to collapse.

****note: The type of pump used in the cementing was a Duplex pump. The pump operates at 50 spm with a volumetric efficiency of 90%. The stroke length is 18 inches with 2.5 inch rods and 6.5 inch liners.***

CHAPTER 3: GAS PROPERTIES DETERMINATION

Determination of Compressibility Factor

There are several gas properties that need to be determined.

- ***Critical pressure:*** that pressure which a gas exerts when in equilibrium with the liquid phase and at critical temperature.
- ***Critical temperature:*** that temperature of a gas above which a gas can not be liquefied by application of pressure alone.

Reduced temperature and reduced pressure are ratios of the actual temperature and pressure. They are written respectively as:

Tr = T/Tc
Pr = P/Pc

The background information in regards to the Standing and Katz gas compressibility chart is based on binary mixtures and saturated hydrocarbon data. The graph correlates pseudo-reduced properties of temperature and pressure. The gas deviation factor can be interpreted as a term by which the pressure must be corrected to account for the departure of a gas form the ideal gas equation as follows:

$$PV = nRT$$

where,

P = pressure
V = volume
N= moles
R = universal gas constant
T = temperature

Natural gases are a mixture of hydrocarbon gases and impurities. The hydrocarbon gases normally found in natural gas are methane, ethane, propane, butanes, pentanes, hexanes, heptanes, octane, and the heavier gases. The impurities found in natural gas include carbon dioxide, nitrogen, hydrogen sulfide, and water vapor.

Natural gas engineers invariably deal with gas mixtures and rarely with single component gases. Since natural gas is a mixture of hydrocarbon compounds and because this mixture is varied in types as well as the relative amount of the compound, the overall physical properties will vary. Physical properties that are most useful in natural gas processing are molecular weight, boiling point, freezing point, density, critical pressure, critical temperature, and specific gravity.

All gases deviate from ideal gas laws under most conditions. Numerous attempts have been made to account for these deviations of real gases from the ideal gas equation of state. One of the most celebrated of these equations is the Van der Waal, but the real gas equation (Boyle's Law) is the most commonly used in practice. Boyle's Law is primarily used for liquid fluids. The real gas equation is a deviation from the ideal gas law using a compressibility coefficient (z-factor):

$$PV = znRT$$

where,

P = pressure
V = volume
N= moles
R = universal gas constant
z = gas compressibility factor
T = temperature

Because gas is compressible, a gas compressibility factor is necessary in Boyle's Law. The Standing and Katz chart is reliable mainly for sweet natural gases, but there are ways to correct for contaminants such as hydrogen sulfide and carbon dioxide.

For this part of the project it was required to calculate and graph the Standing and Katz gas compressibility chart by using the Redlich Kwong Equation of State. The Redlich Kwong EOS uses the theory of cubic equations along with two empirical constants. These types of equations are used to calculate physical properties and vapor-liquid equilibrium of hydrocarbon mixtures. The general form of the Redlich Kwong EOS is as follows:

$$P + \left(\frac{a}{\left(V\,(V + B) * \sqrt{T}\right)} \right)(V - b) = RT$$

where,

P = pressure
V = volume
N= moles
R = universal gas constant
Z = gas compressibility factor
T = temperature

The empirical constants a and b are defined by:

$$a = \Omega_a * \left(\frac{R_2 * T^{2.5}}{P_C} \right)$$

$$\Omega_a = .427480$$

$$b = \Omega_b * \left(\frac{R * T_C}{P_C} \right)$$

$$\Omega_b = .086640$$

By using the two empirical constants and the general formula for Redlich Kwong EOS the following can be derived:

$$f(x_o) = z^3 - z^2 + \frac{P_r}{T_r}\left[\left(\frac{.42748}{T_r^{1.5}}\right) - 0.08664 - .007506 * \frac{P_r}{T_r}\right] * z - .03704 * \frac{P_r^2}{T_r^{3.5}}$$

$$f'(x_o) = 3z^2 - 2z + \frac{P_r}{T_r}\left[\left(\frac{.42748}{T_r^{1.5}}\right) - 0.08664 - .007506 * \frac{P_r}{T_r}\right]$$

Then using the Newton Raphson Iteration Techniques to solve the above equation, the following equation was used to solve for the gas deviation factor.

$$x_1 = x_o - \left(\frac{f(x_o)}{f'(x_o)} \right)$$

By applying an initial guess (x_o) for the gas deviation factor, and substituting it into the derived equation a new gas deviation factor is equated. The absolute value of the difference between the guess and the calculated value is evaluated until it is less than one hundredth of one. If the calculated is within 1/100 then that calculated number becomes your gas deviation factor, but if it does not then that calculated becomes your guess and it is recalculated until it does fall within 1/100.

By using the above equations in a computer program we were able to graph the Standing and Katz gas compressibility chart using visual basic. The program uses nine specific pseudo-reduced temperatures supplied by the instructor. Then, by using the same pseudo-reduced temperature for 150 pseudo-reduced pressures, the program calculates 150 gas compressibility factors. The program repeats the procedure for the other eight pseudo-reduced temperatures. Thereby resulting in one array of pseudo-reduced pressures and nine arrays of gas compressibility factors. Finally, graphing the nine different z-factors vs. the pseudo-reduced pressure array yields the Standing-Katz gas compressibility chart. This depiction of the Standing and Katz plot can be found in the Appendix.

Determination of Viscosity

The coefficient of viscosity is measure of resistance to flow exerted by a fluid. The only accurate method to obtain the viscosity of a gas is to determine it experimentally. The viscosity of a pure gas depends on the temperature and pressure, but for a gas mixture it is also a function of composition of the mixture. The method used to determine the viscosity of this gas for this project was the correlation of Lee, Gonzalez, and Eakian. Graphs were developed to compare the viscosity of gases to the corresponding reservoir pressures. The graphs appear in the appendix portion of this report.

Viscosity = K*0.0001* exp(X*D*^Y)

where,

K = ((9.4+0.02*m)*T^1.5)/(209+19*m + T)
Y = 2.4 – 0.2 * X
X = 3.5 + 986 / T + 0.01 * m
D = (m*P) / (Z *10.7 *T) * (1000/2.205) * (1/30.48^3)
M = Molecular Weight
T = Temperature, R
P = Pressure, psi
Z = Z-factor

Determination of Compressibility

Compressibility of a gas is defined as the fractional change in the volume per unit pressure change.

Cg = Cr / Pc
Cr = Reduced compressibility
Pc = critical pressure

CHAPTER 4: WELL LOG INTERPRETATION

I. Background to Log Interpretation:

First, in order to do well log analysis an individual must have well logs. The idea is to obtain as much information about the formations as possible. The contractor is issued a contract for logging the well, based on the type of logs the company feels necessary to run. The logs were chosen to be run in the two wells were the neutron porosity, the density porosity (density), induction, and the gamma ray. There was also another log run down hole for the Weir formation in the Elkhurst well called the compensated neutron log (CNL).

When the target formations were initially researched by the company the formations had all been documented as having sandstone matrices. However, after interpreting the well logs, it was determined that all the previously known formations were of a limestone matrix. The only sandstone formation that was discovered was the prospect zone (Maxton Sand), and this zone even appears to be limey sandstone. The sandstone formation is only a prospect right now and is being considered for later completion. The matrix densities that indicate how to classify the formations are as follows:

- ***Sandstone: 2.65 g/cc***
- ***Limestone: 2.71 g/cc***
- ***Dolomite: 2.87 g/cc***

Theoretically, each of logs operate as follows:

- The **neutron porosity log** measures the ability to attenuate (slow down) the passage of neutrons through the formation. Attenuating takes place in a couple of different ways, the first is by thermal or epithermal state. This occurs when the temperature changes dramatically. Another way to attenuate the neutrons is when the neutrons elastically collide with a Hydrogen atom, being that they are of almost equal weight.
- The **density porosity log** is a tool that sends gamma rays into the formation and then counts them when they return back at a fixed distance. The gamma rays slow down in certain fluids such as water and gas. In the different fluids, the gamma rays travel at different velocities.

*****note**: The neutron and density porosity logs are run at the same time, and where the two split or go the opposite direction is a very good indication of a gas bearing zone.*

- The **induction log** requires that a current be sent through the borehole fluid into the formation. Often times this is not possible because the borehole fluid is non-conductive (electron flow inhibiting) by nature.

The tool then has to be adjusted so that it is in direct contact with the borehole wall.

- The **gamma ray log** measures the formations natural radioactivity. The three main elements that the tool measures are thorium, uranium, and potassium. The tool measures energy of less than 0.5 and greater than 2.5 millivolts. When this logs spikes in value it is a good indication of a gas-bearing zone.

After interpreting the logs and finding the gas bearing zones, the engineer is able to determine the porosity and gas saturation of the formation in question. After the porosity has been determined, the Initial Gas in Place can be calculated along with the Estimated Ultimate Recovery of the reservoir.

II. Log Interpretation Process:

Almost all oil and gas produced today comes from accumulations in the pore spaces of the reservoir rock, usually sandstone, limestone, or dolomites. The amount of oil or gas contained in a unit volume of the reservoir is the product of its porosity by the hydrocarbon saturation.

In addition to the porosity and the hydrocarbon saturation, the volume of the formation-containing hydrocarbon is needed in order to estimate total reserves and to determine if the accumulation is commercial. Knowledge of the thickness and area of the reservoir is necessary for computation of its volume.

The main petrophysical parameters needed to evaluate a reservoir are porosity, hydrocarbon saturation, thickness, area, and permeability. In addition, the reservoir geometry, formation temperature and pressure, and lithology can play important roles in the evaluation, completion, and production of a reservoir.

Log interpretation is the process by which these measurable parameters are translated into the desired petrophysical parameters of porosity, hydrocarbon saturation producibility, lithology, and mechanical properties.

Since the petrophysical parameters of virgin formation are usually needed, the well logging tool must be able to "see" beyond the casing and cement into the virgin formation, or the interpretation techniques must be able to compensate for this environmental effects.

Determination of Porosity:

Porosity is the pore volume per unit volume of formation; it is the fraction of the total volume of a sample that is occupied by pores or voids. The symbol for porosity is (ϕ). A dense, uniform substance, such as a piece of glass, has almost zero porosity; a sponge, on the other hand, has a very high porosity.

Porosities of subsurface formation can vary widely. Dense carbonates (limestones and dolomites) and evaporites (salt, anhydrite, gypsum) may show practically zero porosity; well-consolidated sandstones may have 10 to 15 % porosity; unconsolidated sands may have 30%, or more, porosity. Shales or clays contain over 40% water-filled porosity, but individually pores usually so small that the rock is impervious to the flow fluids.

Elkhurst Well:

The calculation of porosity for the Elkhurst was based on log interpretation with CNL-density. From this log values were recorded for bulk density, neutron porosity, and compensated neutron porosity. The only way to determine the porosity with this given information was by using the crossplot SW-11 charts (Schlumberger charts book - located in the Appendix section of this report). With the bulk density, neutron porosity, and the matrix of the formation, the porosity could be determine as:

Formation = **WEIR**

Logs**: DUAL INDUCTION (SFL) – CNL DENSITY – GAMMA RAY**

Formation	**ϕN, %**	**ρbulk, g/cc**	**ϕCNL, %**	**Matrix density, g/cc**	**ϕ, %**
Limestone	2.4	2.35	5.0	2.71	12

*** See Appendix for Log interpretation Fig.1*
*** See Appendix for Log Crossplot Fig.2*

Strange Creek Well:

The calculation of porosity for the Strange Creek was also based on log interpretation, but in this well density and neutron porosity logs were run. From these logs we could read values for density porosity and neutron porosity. A different method was used for the interpretation process for this well by going to the crossplot graph Sw-11 or using the porosity equation located in the appendix. With the density porosity, neutron porosity, and the matrix of the formation, the porosity can then be determined by the use of the porosity equation.

Logs**: DUAL INDUCTION (SPCD) – DENSITY – GAMMA RAY**

Formation	**ϕD, %**	**ϕN, %**	**Rw, Ωm**	**ρb, g/cc**	**ρma, g/cc**	**RID, Ωm**	**ϕ, %**
Blue Monday	15.5	3.5	0.05	2.54	2.83	54	10.5
Big lime	8.5	1	0.05	2.68	2.83	325	5
Maxton	11	3	0.05	2.59	2.68	85	7

*** See Appendix for Log Interpretation Fig.3, and Fig.4*
*** See Appendix for crossplot graph Fig.5, Fig.6, and Fig .7*

Determination of Gas Saturation:

The saturation of the formation is the fraction of its pore volume occupied by the fluid considered. Gas saturation, then, is the fraction or percentage of the pore volume that contains formation gas. If only water exists, a formation is 100% water saturated. The symbol for gas saturation is (Sg), water saturation (Sw), and oil saturation (So). The summation of all saturation in a given formation rock must be equivalent to 100%.

CALCULATION OF GAS SATURATION FOR WELL 1 AND WELL 2

The way that both well determined the gas saturation was by using the crossplot for saturation, Sw-11 (Schlumberger charts book). This plot is used through the combination of the density-neutron measurements or density resistivity measurements. To use, enter the chart vertically from the intersection of the apparent bulk density values. The intersection of this line with either the neutron porosity (Corrected for lithology) or the Rt/Rw ratio defines actual gas saturation.

Elkhurst Well:

Formation	ϕN, %	ρbulk, g/cc	ϕCNL, %	Matrix density, g/cc	Sg, %
Limestone	2.4	2.35	5.0	2.71	64

See Appendix for crossplot graph. Fig.2

Strange Creek Well:

Formation	ϕD, %	ϕN, %	Rw, Ωm	ρb, g/cc	ρma, g/cc	RID, Ωm	Sg, %
Blue Monday	15.5	3.5	0.05	2.54	2.83	54	65
Big lime	8.5	1	0.05	2.68	2.83	325	65
Maxton	11	3	0.05	2.59	2.68	85	25

*** See Appendix for crossplot graph Fig. 5, Fig.6, and Fig. 7*

Initial Gas in Place

Initial gas in place (IGIP) deals with the quantity of gas in the reservoir at the time that the reservoir is initially tapped into when explored. Although the calculation process is not extremely complex, this calculation is very important to the engineer. The IGIP estimate gives knowledge of exactly how much production may come out of a given well, thus helping in the decision for how long to develop the reservoir.

In this particular project, the estimate was to be given on a per acre basis. The gas gravity and temperature of the formations also had to be known in order to determine certain parameters in the calculation process. The equations used to calculate the IGIP are shown in the appendix section.

In order to calculate the initial gas formation volume factor, the z-factor had to be calculated. This is where the assumption for gas gravity came into play. It was assumed that the gas being produced from all of the reservoirs had a gas gravity of 0.7, a mean annual surface temperature of 75 degrees Fahrenheit and a temperature gradient of 0.75 degrees Fahrenheit per every 100-foot interval subsurface. Formation temperature needed to be calculated so that the Initial gas in place could be calculated. The equations for calculating formation temperature are located in the appendix section. The z-factor for this calculation comes from the Redlich Kwong approach discussed in the Gas Properties chapter of this report.

Due to the long calculation process described above, a small program was developed to calculate the IGIP using Microsoft Excel. The final values for initial gas in place are listed below in tabular form. The actual program was written in Microsoft Excel, and can be viewed in the Appendix.

Well Name	***Formation Type***	***IGIP (scf/acre)***	***totals***
Elkhurst	Weir	**1,623,059**	***1,623,059***
Strange Creek	Big Lime	**3,813,390**	
Strange Creek	Blue Monday	**2,373,226**	***6,186,616***
Strange Creek	Maxton Sand (prospect)	**299,801**	***6,486,417***

A hard copy of the spreadsheet program can be found in the Appendix.

Estimated Ultimate Recovery

The calculation of the estimated ultimate recovery is very simple once the initial gas in place has been calculated. The only difficulty involved lies with the assumption of the recovery factor to be used. The recovery factor is a fraction dealing with the amount of the gas that will be produced from the calculated initial gas in place. In order to determine the estimated ultimate recovery, the recovery factor is simply multiplied to the value for the initial gas in place. Therefore, the estimated ultimate recovery is given in standard cubic feet (scf) per acre, just as the IGIP. The program developed for calculating the IGIP also calculates the estimated ultimate recovery. Reference back to the previous section on calculating the initial gas in place to see this spreadsheet program.

The calculated values for the estimated ultimate recovery of the reservoirs, and the overall values for each of the wells are displayed below in tabular form.

~~Well Name~~	~~Formation Type~~	~~IGIP (scf)~~	~~EUR~~	~~EUR totals~~
Elkhurst	Weir	**1,623,059**	**1,379,600**	***1,379,600***
Strange Creek	Big Lime	**3,813,390**	**3,241,382**	
Strange Creek	Blue Monday	**2,373,226**	**2,017,242**	***5,258,624***
Strange Creek	Maxton Sand (prospect)	**299,801**	**254,831**	***5,513,454***

CHAPTER 5: BUILD-UP DATA ANALYSIS

I. Introduction to Build-Up Analysis:

Approach (Interpretation and Calculation Process):

The z-factor, compressibility, and viscosity were all calculated using the program developed in PNGE 270 project number one taught by Doctor Kashy Amimian. That first project was incorporated into this project by adding several features. The actual approach to interpreting and calculating (permeability, skin, and effective wellbore storage) the data was fairly simple. Three different methods for the analysis were examined. These three methods were as follows:

1.) Pseudo-Pressure and Time Approach
2.) Pseudo-Pressure and Pseudo-Time Approach
3.) Adjusted Pressure and Adjusted Time Approach

The approach used in each of these methods was as follows:

1.) *Pseudo-Pressure and Actual Time Approach:*
A computer program was written to calculate all of the necessary parameters for the following graphs and correlations. The parameters that needed to be calculated were pseudo pressure and horner time.

b.) A plot of Pseudo-Pressure versus Pressure was developed.
c.) A log-log plot of the difference in Pseudo-Pressure versus the difference in Actual Time was developed in order to determine the start of the semi-log straight line to be evaluated on the Horner plot.
d.) A semi-log plot of Pseudo-Pressure versus Horner Time was developed, and the slope (m) of the semi-log straight line to be evaluated was recorded in psi^2 per centipoise per log cycle. A value for m(P*) was also extrapolated from this particular plot. This value was then corresponded to a value for e.) P* from the Pseudo-Pressure versus Pressure plot discussed in step (b.)

A value for permeability (k) was then calculated using the slope.
Then the skin factor (S′) was calculated.

2.) Pseudo-Pressure and Pseudo-Time Approach:
A computer program was written to calculate all of the necessary parameters for the following graphs and correlations. The parameters that were calculated were pseudo pressure, pseudo time, pseudo production time, and pseudo horner time.

a.) A log-log plot of the difference in Pseudo-Pressure versus the difference in Pseudo-Time was developed in order to determine the start of the semi-log straight line to be evaluated on the Horner plot.

b.) A semi-log plot of Pseudo-Pressure versus Pseudo Horner Time was developed, and the slope (m) of the semi-log straight line to be evaluated was recorded in psi^2 per centipoise per log cycle. A value for m(P*) was also extrapolated from this particular plot. This value was never actually used in the calculation process.
c.) A value for permeability (k) was then calculated using the slope. This was calculated through the use of the following equation:
d.) The skin factor (S′) was calculated using the following equation:
e.) The effective wellbore storage factor was calculated using the following equation:

A value for permeability (k) was then calculated using the slope.
Then the skin factor (S′) was calculated.

3.) *Adjusted Pressure and Adjusted Time Approach:*

A computer program was written to calculate all of the necessary parameters for the following graphs and correlations. The parameters that were calculated were adjusted pressure, adjusted time, adjusted production time, and adjusted horner time.

a.) A log-log plot of the difference in Adjusted Pressure versus the difference in Adjusted Time was developed in order to determine the start of the semi-log straight line to be evaluated on the Horner plot.
b.) A semi-log plot of Adjusted Pressure versus Adjusted Horner Time was developed, and the slope (m) of the semi-log straight line to be evaluated was recorded in psi per log cycle. A value for P_A* was also extrapolated from this particular plot. This value was never actually used in the calculation process.
c.) A value for permeability (k) was then calculated using the slope.
d.) Then the skin factor (S′) was calculated.

A value for permeability (k) was then calculated using the slope.
Then the skin factor (S′) was calculated.

II. Calculation of Permeability and Skin:

In this project a program was developed to calculate various parameters associated with the interpretation of pressure build-up data for natural gas wells. The program receives the following input parameters:

- formation temperature
- initial pressure
- gas gravity
- flow rate
- formation thickness
- porosity
- well radius
- a data file of the build-up pressures with the corresponding shut-in times

The program calculates the pseudo properties of both the pressure and the time, in conjunction with the adjusted values for each pseudo property. All of this data is displayed in the program, and in a corresponding data file (“a:results.txt”). The data file is then opened in Microsoft Excel, where the build-up data can be graphically displayed, and easily manipulated, on both log-log plots and Horner plots for the each of the three scenarios to be evaluated. The three scenarios are as follows:

- Pseudo-Pressure and Time
- Pseudo-Pressure and Pseudo-Time
- Adjusted Pressure and Adjusted Time

After the data is analyzed and the interpreted values are recorded, the program contains an additional form to input the recorded values and calculate the permeability, skin, and effective wellbore storage constant.

In order to calculate the permeability, skin factor, and the dimensionless effective wellbore storage constant for the three different approaches, some other parameters had to be evaluated. These parameters included the z-factor, viscosity, and compressibility at each of the pressures from one psi to the initial reservoir pressure (one psi pressure steps were evaluated in the program). These properties were equated using the methods described in the Gas Properties section of the report.

The permeability, skin factor, and dimensionless effective wellbore storage factor for each of the approaches was then calculated. The equations used for each of the three approaches are listed below:

1.) Pseudo-Pressure and Actual Time Approach:

$$k\ (mD) = (1637*q*T)/(m*h)$$

where,

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
m = slope of line (psi^2/cp/cycle)
h = formation thickness (feet)

$$S' = 1.151 * \{[(m(P_{wf}) - m(P_{ws})) / m] - [\log((k*\Delta t)/(\phi*\mu*c_t*(r_w^2)))] + 3.23\}$$

where,

$m(P_{wf})$ = flowing pressure of the well (psi^2/cp)
$m(P_{ws})$ = static pressure of the well (psi^2/cp)
m = slope of line (psi^2/cp/cycle)
ϕ = porosity (fraction)
μ = viscosity (cp)
c_t = compressibility (psi^{-1})
r_w = wellbore radius (feet)
Δt = SI time at $m(P_{ws})$
k = permeability (mD)

2.) Pseudo-Pressure and Pseudo-Time Approach:

$k = (1637*q*T)/(m*h)$

where,

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
m = slope of line ($psi^2/cp/cycle$)
h = formation thickness (feet)

$S' = 1.151 * \{[(m(P_{wf}) - m(P_{ws})) / m] – [\log((k*\Delta t_{ap})/(\phi *(r_w^2)))] + 3.23 + [\log(t_{pap}+\Delta t_{ap}) / (t_{pap})] \}$

where,

$m(P_{wf})$ = flowing pressure of the well (psi^2/cp)
r_w = wellbore radius (feet)
$m(P_{ws})$ = static pressure of the well (psi^2/cp)
m = slope of line ($psi^2/cp/cycle$)
k = permeability (mD)
t_{pap} = Pseudo Production Time
Δt_{ap} = Pseudo-Time at $m(P_{ws})$
ϕ = porosity (fraction)

$C_{Deff} = \{[13.26*q*P_{sc}*T]/[\phi*h*(r_w^2)*T_{sc}]\} * [\Delta t_{ap}/\Delta m(P)]$

where,

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
T_{sc} = temperature at standard conditions (520 degrees Rankin)
h = formation thickness (feet)
r_w = wellbore radius (feet)
P_{sc} = pressure at standard conditions (14.7 psia)
ϕ = porosity (fraction)
Δt_{ap} = Pseudo-Time at $\Delta m(P)$
$\Delta m(P)$ = difference in Pseudo-Pressure at Δt_{ap}

3.) Adjusted Pressure and Adjusted Time Approach:

$k = -162.6 * [(q) * (B_{gavg}) * (\mu_{avg})] / (m * h)$

where,

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
m = slope of line ($psi^2/cp/cycle$)
h = formation thickness (feet)
μ_{avg} = viscosity at average pressure (cp)
B_{gavg} = average gas formation volume factor (RB/Mscf)

$S' = 1.151 * \{[(m(P_{wf}) - m(P_{ws})) / m] – [\log((k*\Delta t_A)/(\phi*\mu_{avg}*c_{tavg}*(r_w^2)))] + 3.23 + [\log(t_{pA}+\Delta t_A) / (t_{pA})] \}$

where,

$m(P_{wf})$ = flowing pressure of the well (psi^2/cp)
$m(P_{ws})$ = static pressure of the well (psi^2/cp)
m = slope of line ($psi^2/cp/cycle$)
k = permeability (mD)
t_{pA} = Pseudo Production Time
Δt_A = Pseudo-Time at $m(P_{ws})$
ϕ = porosity (fraction)
μ_{avg} = viscosity at average pressure (cp)
r_w = wellbore radius (feet)
c_{tavg} = compressibility at average pressure (psi^{-1})

The results of the runs can be viewed in the Appendix section of this report.

The final calculated values for the permeability, skin factor, effective wellbore storage in the formations located in the two wells are as follows:

Well	Formation	k (mD)	Skin	Wellbore Storage
Elkhurst	Weir	0.4	10.28	3,914
Strange Creek	Big Injun	1.27	13.48	1,777
	Blue Monday	1.59	-5.68	298,191

These calculated values yielded that the Weir and Big Injun formations are candidates for an acid-fracturing job. The Blue Monday was already stimulated (natural fractures). As a rule of thumb a natural fracture is said to be occurred when an indication of the skin factor is greater than –4 that is why that such a big number for effective wellbore storage is obtained, and also no damage appears to be present at the formation face. The results of the three runs are displayed in the appendix chapter of this report.

CHAPTER 6: FLOW RATE OPTIMIZATION

I. Background for Flow Optimization:

The pressures associated with gas deliverability are as follows: reservoir pressure (P), flowing bottom hole pressure (Pwf), and wellhead pressure (Ptf). These pressures can be related to one another through different methods. For example, the line source solution can be used in relating Pwf to the reservoir pressure, as can the deliverability equations. There are three methods of relating Pwf to Ptf. They are the average temperature and deviation factor approach, the Sukkar and Cornell approach, and the Cullender and Smith approach. For any of the relationships, the flow rate always plays an important role in the calculation process.

Sukkar and Cornell published tabular data for solving Pwf calculation problems within the reduced temperature-reduced pressure range of $1.5 < Tpr < 1.7$; $1 < Ppr < 12$. This method only applied for vertical wells, and the assumption made in this method include:

- Steady state flow
- Single-phase flow
- Change in kinetic energy
- Temperature is constant
- Friction is constant

All equations used in the Sukkar and Cornell are based on the assumption that the temperature is constant at some average value. The Cullender and Smith method also depends on the following assumptions:

- Steady-state flow
- Single-phase gas stream
- Change in kinetic energy

The ability of the reservoir to deliver a certain quantity of gas depends on the inflow and outflow performance relationship associated with the reservoir. For the determination of a constant flow rate to produce the reservoir over the contract period, the following procedure was utilized.

- An abandonment pressure was determined based on the separator pressure.
- The initial reservoir pressure was then decremented, and the flowing bottom hole pressure (Pwf) at each pressure step (1 psi) was calculated using the deliverability equations found in the Appendix.
- For each of the calculated flowing bottom hole pressures, the wellhead pressure at that point was also calculated using the average z-factor, average temperature approach. Though this approach is not as accurate, it is the most practical based on its simplicity. The equation for this approach is also located in the Appendix section of the report.

II. Calculation of Optimum Flow Rate:

A program was developed to equate an optimum flow rate using the following procedure:

1.) The flow rate (q) was initially 0.01 Mcfd.
2.) The initial pressure was then decremented by 1/1000 from the initial pressure.
3.) At each of these pressures, the flow rate (q) was used to equate the sandface pressure from the deliverability equation
4.) The wellhead pressure was then calculated at that sandface pressure using the average z-factor, average temperature approach (Appendix section).
5.) The gas produced ($G_p = \frac{G \times (B_g - B_{gi})}{B_g}$) was then determined for each of the decremented pressures.
6.) The time period ($t = G_p/q$) associated with the decremented pressure was then equated for that constant rate.
7.) If the final time period for producing at that constant rate was not equivalent to the seven year contract life, then the flow rate was incremented by 0.01 Mcfd and the procedure was repeated.

The user then has the option to view the results of the program on separate forms, or to view the results in output files ("a:weir.txt", "a:biginj.txt", and "a:bluem.txt"). The forms to view the results are pictured below.

The results of the program are displayed below in tabular form:

Well	Formation	Flow Rate (Mcfd)	Time (yr)	Time (days)
Strange Creek	Blue Monday	0.79	7.01	2558.65
	Big Injun	1.41	7.01	2558.65
Elkhurst	Weir	0.59	7.01	2558.65

CHAPTER 7: ECONOMICS

I. Background for Monte Carlo Simulation and Distribution Types:

YARDSTICKS THAT INCORPORATE THE TIME VALUE OF MONEY

The current trend is toward yardsticks that incorporate the time value of money. However, prior to 1950 this was not the case. During the 1950's petroleum companies and public utilities began to use yardsticks that incorporated the time value of money, and surveys subsequent to 1960 indicate an increasing use of the discounted yardsticks and a decreasing use of the accounting rate of return and payback methods. The following paragraphs will present some of the yardsticks that incorporate the time value of money.

Net Present Value

Method of Calculation

The net present value of an investment is calculated by discounting the future net cash flows to time zero and summing them. In its simplest form, the equation for the net present value can be expressed as:

$$\sum = NCF_j / (1 + I)\ \text{^}j$$

where:

L = project life
NCF_j = net cash flow for period j
NPV = net present value discounted at rate i

The above equation assumes that investments subsequent to time zero are treated as cash out and are included in the annual NCF calculation. It also assumes that all net cash flows occur at the end of the period.

If money has a time value of X % to me, would I rather invest a certain amount of money in exchange for an estimated future cash stream, or keep my money? In order to make this decision, all cash flows need to be measured at some common time. Usually this is time 0. By doing this, one can operated the initial investment at time 0 with the present value (discounted at your time value of money) of the future net cash flows. If the present value of the future net cash flows is greater than the time 0 investment, then you would prefer the investment and future cash stream over keeping your money. In the case where your time 0 investment is equal to the present value of the future net cash flow stream, you would be indifferent. When the time 0 investment is greater than the discounted future net cash flow stream, you would prefer to keep your money and look for another opportunity.

Discounted Cash Flow Rate of Return (DCFROR)

The discounted rate of return (DCFROR) is a widely used yardstick and is the primary measure of investment worth for many firms. The DCFROR yardstick is also called discounted rate of return, effective yield to maturity, return on investment, profitability index, and earning power.

DCFROR is widely used since the yardstick is relatively easy to use; and, unlike the NPV method, does not require that a discount rate be established prior to making the calculation. This apparent advantage however, is the reason why DCFROR has a certain limitations when compared to the NPV method. The limitations and assumptions associated with DCFROR will be discussed in the paragraphs that follow.

Method of Calculation.

The DCFROR on investment is calculated by discounting the cash flow at various discount rates until the sum of the discounted net cash flows equals zero. The calculation of the DCFROR is and trial and error process. In equation form, the DCFROR can be expressed as:

$$\sum = NCF_j / (1 + I) \hat{} j = 0$$

where:

L = project life.

NCF_j = net cash flow for period j.

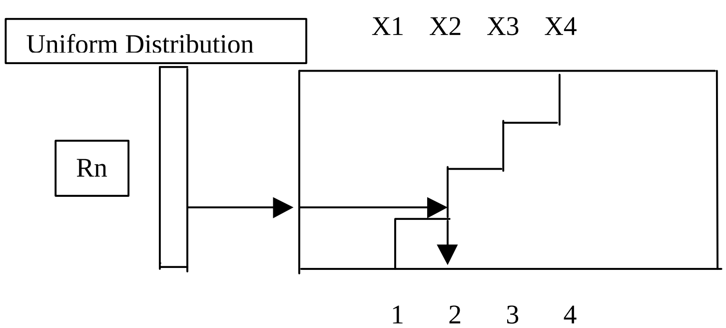
I = DCFROR

The DCFROR has been described in the literature in many ways but what does it really mean? Supposed assumptions about the reinvestment of cash flows and the idea of an internal rate of return seem to cloud the issue. In making an investment, you are simply buying an annuity or series of future cash flow streams. Think of making a bank deposit, equivalent to the amount of your investment, in a bank paying interest at a rate equal to the project's annual net cash flow streams. At the end of the project, your balance in the bank will be zero. Notice that in the analysis, no mention was made concerning what you do with the money that you receive from the bank. Accordingly, there is no assumption about the reinvestment of future cash flows in the DCFROR technique. The technique to be valid, you must invest all the annual net cash flow at the DCFROR or the internal rate of return as it is referred to in the financial literature.

The use of probability and statistical methods in computer software, and many other areas involves the gathering of information, or scientific data. Of course, the gathering of data is nothing new to the engineer. Data is collected, summarized, and stored on a day to day basis in engineering research. However, there is a profound distinction between collection and inferential statistics. In this project three methods were used to distribute the ranges of uncertainty. These methods include: Uniform Distribution, Triangular distribution, and discrete distribution.

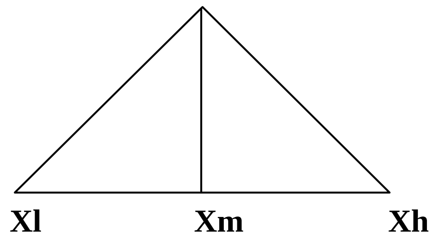
Discrete Uniform Distribution

The simplest of all discrete probability distribution is one in which the random variable assumes each of its values with equal probability. Such a probability distribution is called discrete uniform distribution.



Required Condition	Change of X
0 < = Rn <= P1	X1
P1 < = Rn < = P1 + P2	X2
P1 +P2 < Rn < = P1 +P2 + P3	X3
P1 +P2 + P3 < Rn < = 1	X4

Triangular Distribution:



When $Xl <= X <= Xm$

$$F (x) = (X – Xl / Xm – Xl)^{2} * (Xm – Xl / Xh – Xl)$$

When $Xm <= X <= Xh$

$$F (x) = 1 - (Xh – X / Xh – Xm)^{2} * (Xh – Xm / Xh – Xl)$$

Replacing F (x) by Ramdon Number:

If $Rn <= [(Xm – Xl) / (Xh – Xl)]$

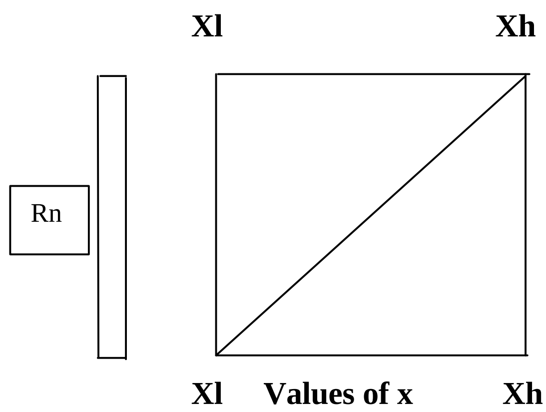
$Rn <= [(Xm – Xl) / (Xh – Xl)]$ then

$X = Xl + [(Xm – Xl) * (Xh – Xl)*Rn]^{0.5}$

If $Rn => [(Xm – Xl) / (Xh – Xl)]$ then

$X = Xh - [(Xh – Xm) * (Xh – Xl)*Rn]^{0.5}$

Uniform Distribution:



Use when upper and lower limits of the range of the variable can be specified, and when any of the values between these limits are as likely to occur as any other value.

$F(x) = X - Xl / Xh - Xl$

Replacing F(x) with Rn, the uniform distributed number and solving for x:

$X = Xl + Rn^* (Xh - Xl)$

II. Economic Results Based on Probability:

A program was developed to simulate various values for DCFROR based on the probability of the reservoir characteristics associated with our three formations. The various distribution types were as follows:

- Permeability Range of 0.1 to 2 mD
- Uniform thickness distribution (± 5 feet)
- Triangular porosity distribution (± 5%)
- Triangular Sg distribution (± 5%)
- Discrete production distribution (scenarios were given by the professor)

The program actually generated a series of random numbers, and corresponded each random number to a value within the range for each property. The distribution was then evaluated, and the initial gas in place was calculated based on the reservoir parameters. Using the production scenario determined for the random number (discrete distribution), the interest rate was incremented until a rate was reached where the Net Present Worth of the project was zero (DCFROR). The frequency of occurrence for each DCFROR over the series of random values was tabulated and plotted versus the DCFROR values. This yielded the most probable value for DCFROR for the project. The forms for the program along with the graphical representations of the distribution types, and the DCFROR results can be found in the Appendix.

The final results yielded that the Weir and Blue Monday each had a DCFROR of 0.09 and that the DCFROR of the Big Injun was 0.15. Based on these results, it is determined that the Strange Creek formation will pay off the fastest and yield the most profit. Theoretically, the two formations in the Strange Creek well have higher or equivalent values for DCFROR. Therefore, this well will be the most profitable.

FINAL RECOMMENDATIONS

Throughout this project, the Strange Creek well had appeared to be the investment of choice. From the log interpretation process, it was evident that the well would probably produce a lot more gas based on the fact that two producing zones appeared to be present. Furthermore, upon the calculation of the gas in place per acre, the Strange Creek appeared to be the well of choice, containing a significantly larger amount of gas, approximately 4,000,000 scf/acre more than the Elkhurst.

Upon the analysis of the build-up data, it was even more apparent that the Strange Creek would probably be the well of choice. The analysis revealed fairly high skin factors for both the Big Injun (Strange Creek well) and Weir (Elkhurst well) formations. However, the other zone located in the Strange Creek yielded a skin factor of around negative five. This value indicated that natural fractures were present and that a stimulation job for this formation would not be necessary. Therefore, the cost for a stimulation job in both wells would be approximately the same. Based on the fact that the Strange Creek would be more productive, it would also allow for the quickest payoff associated with the stimulation treatment, and was still expected to be the best investment.

In the optimization of the flow rate, the Strange Creek blew the Elkhurst out of the water. It had the two highest flow rates of the three formations evaluated. This almost sealed the Strange Creek well as being the best investment. If the predicted flow rates are higher, then it is also most likely that the greatest revenue will occur for that particular well containing the most productive formation. In this case, the Strange Creek contained the TWO most productive flow rates. The Strange Creek once again appeared to be the investment of choice.

The final and main test to decide whether the Strange Creek would be the better investment was in the determination of the Discount Cash Flow Rate of Return. The Strange Creek well yielded higher values for DCFROR. Therefore, the Strange Creek well will be the best investment. It is determined by our company that the Strange Creek venture is the project of choice.

REFERENCES

"*Field Data Hand Book*"; by Principal Dowell Offices; published by Schlumberger

"*Basic Petroluem Geology*"; by Peter K. Link in the years 1996; published by Oil and Gas Consultants International Inc.

"*Applied Drilling Engineering*"; by Adam T. Bourgoyne, Jr., Martin E. Chenevert; Keith K. Millhiem, F.S. Young, Jr.; published by Society of Petroleum Engineers; 1991

"*State of West Virginia Department of Mines Oil and Gas Division*"; published by Allegheny Land Mineral Co.; 1982

"*Standard Handbook of Petroleum and Natural Gas Engineering*"; by William C. Lyons Ph.D., P.E.; published by Gulf Publishing Company; 1996

*"Schlumberger Log Interpretation Charts"***;** by Schlumberger Wireline Company; 1996

"*Schlumberger Cased Hole Log Interpretation Principles/Applications***"**; by Schlumberger Wireline Company; 1989

"*Petroleum and Natural gas Engineering 235 (book of notes*)**";** by Professor Sam Ameri; West Virginia University; Teacher's Press; 1998

"*Theory, Measurement, and Interpretation of Well Logs*"; by Zaki Bassiouni; Society for Petroleum Engineers Textbook (Series Vol. 4); Society of Petroleum Engineers, Inc.; 1994

"*Natural gas Production Engineering****";*** by Chi U. Ikoku; Kreiger Publishing Company; 1992

"Petrolum and Natural gas Engineering 271: Natural Gas Production and Storage Engineering"; Kashy Aminian, Ph.D.; West Virginia University; published by Teacher's Press; 1999

APPENDIX

CASING

Step1 (well 1 - ELKHURST)

Surface Casing clearance = Bit size – OD coupling
= 12.25 – 9.625 = 2.625 in
Production Casing clearance = Bit size – OD coupling
= 4.5 – 5.0 = 1.75 in

Step 2 (DEPTH SELECTION)

The selection of the number of casing strings and their respective setting depths generally is based on a consideration of the pore pressure gradient and fracture gradient of the formation to be penetrated. The pore pressure and the fracture pressure gradient are expressed in equivalent density, and are plotted Vs depth. Here are the two graphs that estimates how many casing are needed and their respective depths.

Step 3 (LOAD CONSIDERATIONS- ELKHURST)

TENSION:

For a casing: 8-5/8” H-40 28.00 lb/ft (Round thread joint)

Pipe body yield strength = 318,000 lbf

Joint strength = 233,000 lbf

318,000 lbf /2 = 159,000 lbf

233,000 lbf /2 = 116,500 lbf

Weight of the casing @ 830’ = 28.00 lb/ft * 830’ = 23,240 lbf

Failure of yield strength = 159,000 lbf – 23,240 lbf = 135,760 lbf

Safety factor = 159,000 lbf / 23,2400 lbf = 6.8*

Failure for joint rupture = 116,500 lbf / 23,240 lbf = 93,260 lbf

Safety factor = 116,500 lbf / 29,880 lbf = 5.0*

* If safety factor is greater than 1.6 then the string is OK

For a casing: <u>4-1/2 H-40 9.5 lb/ft (Round thread joint)</u>

Pipe body yield strength = 111,000 lbf

Joint strength = 77,000 lbf

111,000 lbf /2 = 55,500 lbf

77,000 lbf /2 = 38,500 lbf

Weight of the casing @ 2394' = 9.5 lb/ft * 2394' = 22,743 lbf

Failure of yield strength = 55,500 lbf – 22,743 lbf = 32,757 lbf

Safety factor = 55,500 lbf / 22,743 lbf = 2.4*

Failure for joint rupture = 38,500 lbf - 22,743 lbf = 15,757 lbf

Safety factor = 38,500 lbf / 22,743 lbf = 1.6*

* If safety factor is greater or equal than 1.6 then the string is OK

BURST PRESSURE:

Pf = 0.052*(13.1)*830' = 610 psia   [Fracture pressure]

The surface casing pressure for the design loading conditions is:

610 – 0.05*(830') = 570 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(830') = 380 psia

The pressure differential that tends to burst the casing is 570 psia at the surface and 230 psia (610 –380) at the casing seat.multiplying this pressure by a safty factor of 1.1 yields a burst design load of 630 psia at the surface and 255 psia at the casing seat.

OD,in	Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
8-5/8	H-40	28.00	2,470	630

COLLAPSE PRESSURE:

The design for collapse is based on the density of mud in the hole when the casing is being run, according to the graph in step 2, you can select the mud density and the

external pressure at 830' is: (assume water in the hole, 8.33 lb/gal + safety factor of 0.6 lb/gal)

$$0.052(8.9)(830') = 385 \text{ psia}$$

The internal pressure for the collapse –design load is controlled by the maximum loss in fluid level that could occur if a severe lost of circulation problem is encountered. The way this is designed by assuming the worst
Scenario where a lost of circulation zone unexpectedly is encountered near the depth of the next casing seat The differential pressure that tends to collapse the casing is zero at the surface Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 420 at 830'.

OD,in	Grade	Weight, lb/ft	Collapse resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**1,610**	**420**

PRODUCTION CASING DESIGN FOR WELL 1: 4-1/2 H-40 9.5 lb/ft @ 2394'

For step 1 and 2 they were previously done, and for step 3 (Tension) was also done in the first part.

BURST PRESSURE:

Pf = 0.052*(14)*2394' = 1670 psia [Fracture pressure]

The surface casing pressure for the design loading conditions is:

1670' – 0.05*(2394') = 1555 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(2394') = 1070 psia

The pressure differential that tends to burst the casing is 1555 psia at the surface and 600 psia (1670 –1070') at the casing seat multiplying this pressure by a safety factor of 1.1, yields a burst design load of 1710 psia at the surface and 660 psia at the casing seat.

OD,in	Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**3,190**	**1070**

COLLAPSE PRESSURE:

The design for collapse is based on the density of mud in the hole when the casing is being run, according to the graph in step 2, you can select the mud density and the external pressure at 2394' is: (assume water in the hole, 8.33 lb/gal + safety factor of 0.6 lb/gal)

0.052(8.9)(2394') = 1060 psia

The internal pressure for the collapse –design load is controlled by the maximum loss in fluid level that could occur if a severe lost of circulation problem is encountered. The way this is designed by assuming the worst Scenario where a lost of circulation zone unexpectedly is encountered near the depth of the next casing seat.

The differential pressure that tends to collapse the casing is zero at the surface Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 1160 at 2294'.

OD,in	Grade	Weight, lb/ft	Collapse resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**2,760**	**1160**

<u>Step1 (WELL 2 STRANGE CREEK)</u>

Surface Casing clearance = Bit size – OD coupling
= 8.75 – 7.656 = 1.094 in
Production Casing clearance = Bit size – OD coupling
= 4.5 – 5.0 = 1.75 in

Step 2 (DEPTH SELECTION)

The selection of the number of casing strings and their respective setting depths generally is based on a consideration of the pore pressure gradient and fracture gradient of the formation to be penetrated. The pore pressure and the fracture pressure gradient are

expressed in equivalent density, and are plotted Vs depth. Here are the two graphs that estimates how many casing are needed and their respective depths.

Step 3 (LOAD CONSIDERATIONS- ELKHURST)

TENSION:

For a casing: 7" H-40 17.00 lb/ft (Round thread joint)

Pipe body yield strength = 196,000 lbf

Joint strength = 122,000 lbf

196,000 lbf /2 = 98,000 lbf

122,000 lbf /2 = 61,000 lbf

Weight of the casing @ 310' = 17.00 lb/ft * 310' = 5,270 lbf

Failure of yield strength = 98,000 lbf – 5,270 lbf = 92,270 lbf

Safety factor = 98,000 lbf / 5,270 lbf = 18.5*

Failure for joint rupture = 61,000 lbf / 23,750 lbf = 37,250 lbf

Safety factor = 61,000 lbf / 23,750 lbf = 11.57*

* If safety factor is greater than 1.6 then the string is OK

For a casing: <u>4-1/2 H-40 9.5 lb/ft (Round thread joint)</u>

Pipe body yield strength = 111,000 lbf

Joint strength = 77,000 lbf

111,000 lbf /2 = 55,500 lbf

77,000 lbf /2 = 38,500 lbf

Weight of the casing @ 1965’ = 9.5 lb/ft * 1965’ = 18,668 lbf

Failure of yield strength = 55,500 lbf – 18,668 lbf = 36,833 lbf

Safety factor = 55,500 lbf / 18,668 lbf = 2.9*

Failure for joint rupture = 38,500 lbf – 18,668 lbf = 19,832 lbf

Safety factor = 38,500 lbf / 18,668 lbf = 2.0*

* If safety factor is greater or equal than 1.6 then the string is OK

BURST PRESSURE:

Pf = 0.052*(13.1)*310’ = 210 psia   [Fracture pressure]

The surface casing pressure for the design loading conditions is:

210 – 0.05*(310’) = 195 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(310’) = 145 psia

The pressure differential that tends to burst the casing is 195 psia at the surface and 65 psia (210 –145) at the casing seat.multiplying this pressure by a safty factor of 1.1 yields a burst design load of 230 psia at the surface and 70 psia at the casing seat.

Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
H-40	**17.00**	**2,310**	**230**

COLLAPSE PRESSURE:

The design for collapse is based on the density of mud in the hole when the casing is being run, according to the graph in step 2, you can select the mud density and the external pressure at 310' is: (assume water in the hole, 8.33 lb/gal + safety factor of 0.6 lb/gal)

$$0.052(8.9)(310') = 145 \text{ psia}$$

The internal pressure for the collapse –design load is controlled by the maximum loss in fluid level that could occur if a severe lost of circulation problem is encountered. The way this is designed by assuming the worst

senario where a lost of circulation zone unexpectedly is encountered near the depth of the next casing seat The differential pressure that tends to collapse the casing is zero at the surface Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 160 at 310'.

OD,in	Grade	Weight, lb/ft	Collapse resistance psia	External pressure + SF, psia
7	H-40	17.00	1,420	160

PRODUCTION CASING DESIGN FOR WELL 2: 4-1/2 H-40 9.5 lb/ft @ 1965'

For step 1 and 2 they were previously done, and for step 3 (Tension) was also done in the first part.

BURST PRESSURE:

Pf = 0.052*(13.6)*1965' = 1380 psia [Fracture pressure]

The surface casing pressure for the design loading conditions is:

1380 – 0.05*(1965') = 1280 psia

At the surface the external pressure is zero, for a normal formation pore pressure of 0.465 psi/ft, the external pressure at the casing seat is:

(0.465)(1965') = 900 psia

The pressure differential that tends to burst the casing is 1280 psia at the surface and 480 psia (1380–900’) at the casing seat multiplying this pressure by a safety factor of 1.1, yields a burst design load of 1500 psia at the surface and 530 psia at the casing seat.

OD,in	Grade	Weight, lb/ft	Internal resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**3,190**	**1500**

COLLAPSE PRESSURE:

The design for collapse is based on the density of mud in the hole when the casing is being run, according to the graph in step 2, you can select the mud density and the external pressure at 1965’ is: (assume water in the hole, 8.33 lb/gal + safety factor of 0.6 lb/gal)

0.052(8.9)(1965’) = 900 psia

The internal pressure for the collapse –design load is controlled by the maximum loss in fluid level that could occur if a severe lost of circulation problem is encountered. The way this is designed by assuming the worst Scenario where a lost of circulation zone unexpectedly is encountered near the depth of the next casing seat.

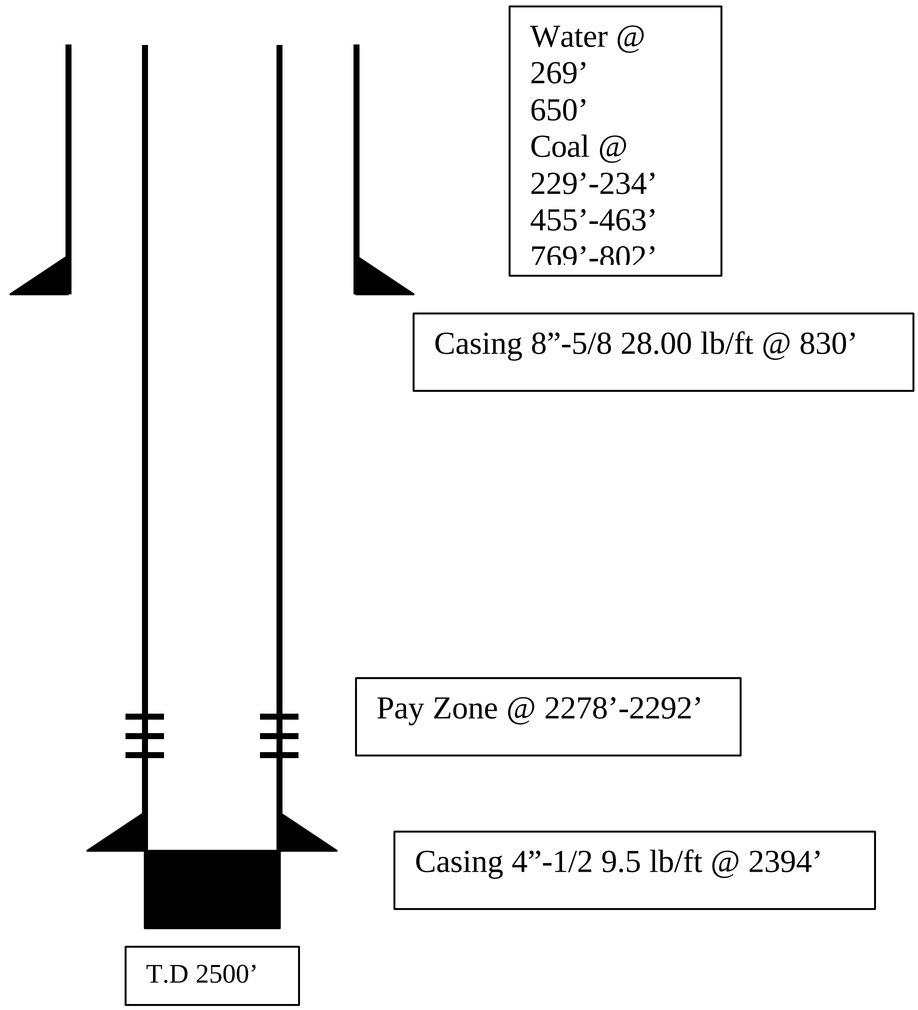
The differential pressure that tends to collapse the casing is zero at the surface

Multiplying this pressure by a safety factor of 1.1 yields a collapse design load of zero at the surface, and 1000 at 1965’.

OD,in	Grade	Weight, lb/ft	Collapse resistance psia	External pressure + SF, psia
4-1/2	**H-40**	**9.5**	**2,760**	**1000**

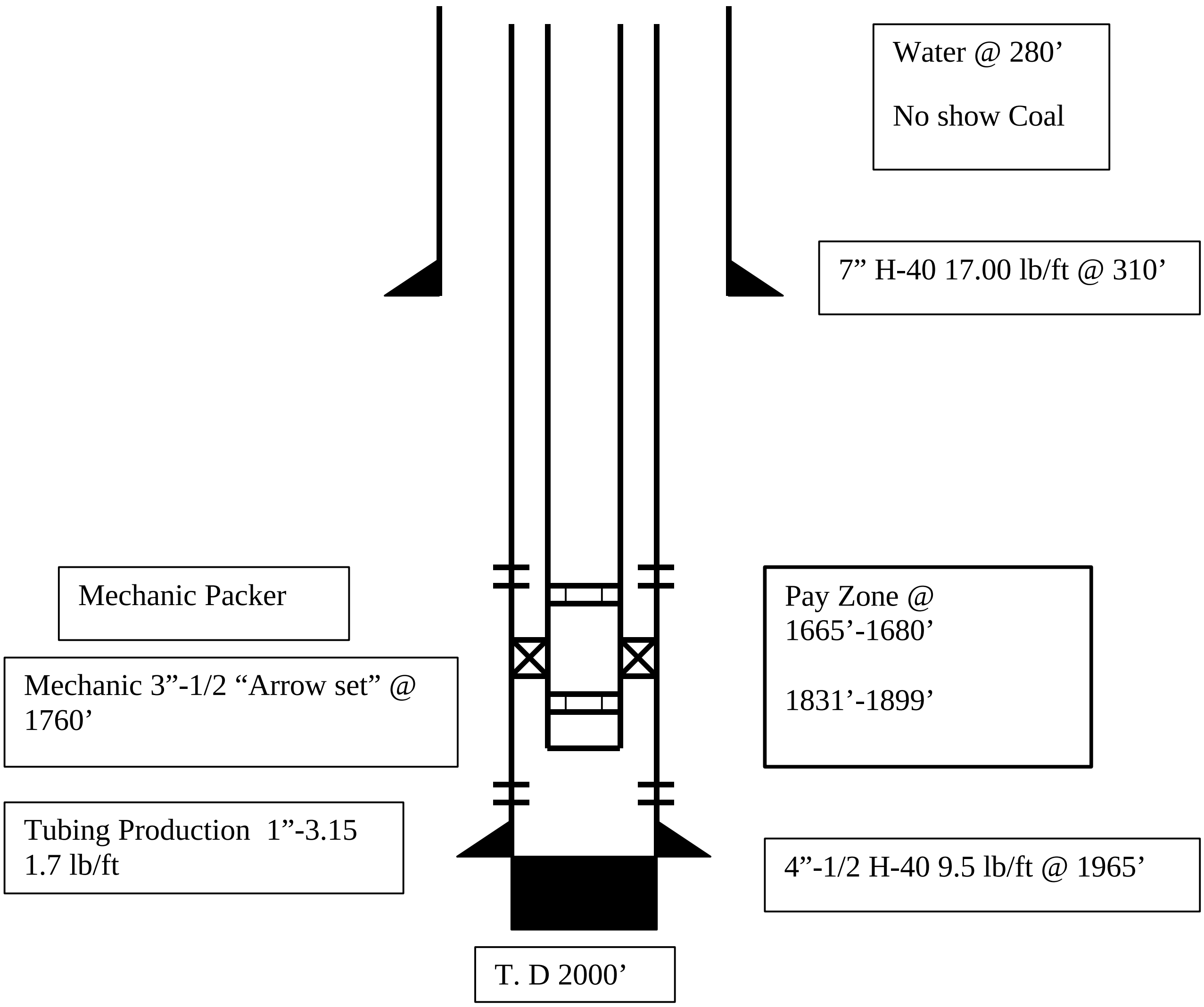
CASING MAKE – UP (WELL 1- ELKHRUST)

WELL: ELKHRUST	DRILLED: JUL –26 – 1982
DEPTH: 2500’	WELL TYPE: GAS



CASING MAKE – UP (WELL 2- STRANGE CREEK)

WELL: STRANGE CREEK	DRILLED: FEB –03 – 1987
DEPTH: 2000’	WELL TYPE: GAS



CEMENTING DESIGN

Elkhurst Cementing Design Calculations

Surface Casing:

Annular Area = [π/4*(11"^{2} – 8.625"^{2}) sq. ft.] / 144" = 0.2542 sq. ft.
Annular Volume = 0.2542 sq. ft.*(830 ft.)*(1.75) = 369 cu. ft.
Sacks = Annular Volume / Cement = (369 cu. ft.) / (1.18 cu. ft./sack) = 313 sacks

Mixing Time = (313 sacks) / [(25 sacks per min.)*60 min./hr] = 0.21 hrs.
Casing Int. Capacity = π/4*[(8.017")^2 / 144] = 0.3506 ft.^{2}
Total Volume = 830 ft.*(0.3506 ft^2) = 291 ft.^{3}
Displacement Time = (291 ft.^{3}) / [50 spm*(1.1523 ft.^{3}/s)*60 min/hr] = 0.064 hrs.
Setup Time = 15 min / (60 min/hr) = 0.25 hrs.
Total Operation Time = 0.21 hrs. + 0.064 hrs. + 0.25 hrs. + 0.5 hrs. = 1.024 hrs.

Production Casing:

Annular Area = [π/4*(6.75"^{2} – 4.5"^{2}) sq. ft.] / 144" = 0.1381 sq. ft.
Annular Volume = 0.1381 sq. ft.*(700 ft.)*(1.75) = 169 cu. ft.
Sacks = Annular Volume / Cement = (169 cu. ft.) / (1.18 cu. ft./sack) = 143 sacks

Mixing Time = (143 sacks) / [(25 sacks per min.)*60 min./hr] = 0.095 hrs.
Casing Int. Capacity = π/4*[(4.090")^2 / 144] = 0.0912 ft.^{2}
Total Volume = 2500 ft.*(0.0912 ft^2) = 228 ft.^{3}
Displacement Time = (228 ft.^{3}) / [50 spm*(1.1523 ft.^{3}/s)*60 min/hr] = 0.066 hrs.
Setup Time = 15 min / (60 min/hr) = 0.25 hrs.
Total Operation Time = 0.095 hrs. + 0.066 hrs. + 0.25 hrs. + 0.5 hrs. = 0.911 hrs.

Total Sacks Cement = 313 + 143 = 456 sacks
Total Time Required = 1.024 + 0.911 = 1.935 hours

*****The pump used for this cement job is a duplex pump operating at 50 spm with a 90% volumetric efficiency. The pump has an 18 inch stroke, and consists of 2.5 inch rods and a 6.5 inch liner. The rate is equivalent to 1.1523 cubic feet per second.***

Strange Creek Cementing Design Calculations

Surface Casing:

Annular Area = [π/4*(8.625"^{2} – 7"^{2}) sq. ft.] / 144" = 0.1385 sq. ft.
Annular Volume = 0.1385 sq. ft.*(310 ft.)*(1.75) = 75 cu. ft.
Sacks = Annular Volume / Cement = (75 cu. ft.) / (1.18 cu. ft./sack) = 64 sacks

Mixing Time = (64 sacks) / [(25 sacks per min.)*60 min./hr] = 0.043 hrs.

Casing Int. Capacity = $\pi/4*[(6.538")^2 / 144] = 0.2331\ ft.^2$
Total Volume = 310 ft.*(0.2331 ft^2) = 72 $ft.^3$
Displacement Time = (72 $ft.^3$) / [50 spm*(1.1523 $ft.^3$/s)*60 min/hr] = 0.021 hrs.
Setup Time = 15 min / (60 min/hr) = 0.25 hrs.
Total Operation Time = 0.043 hrs. + 0.021 hrs. + 0.25 hrs. + 0.5 hrs. = 0.814 hrs.

Production Casing:

Annular Area = $[\pi/4*(6.00"^2 - 4.5"^2)$ sq. ft.] / 144" = 0.0859 sq. ft.
Annular Volume = 0.0859 sq. ft.*(700 ft.)*(1.75) = 105 cu. ft.
Sacks = Annular Volume / Cement = (105 cu. ft.) / (1.18 cu. ft./sack) = 90 sacks

Mixing Time = (90 sacks) / [(25 sacks per min.)*60 min./hr] = 0.06 hrs.
Casing Int. Capacity = $\pi/4*[(4.090")^2 / 144] = 0.0912\ ft.^2$
Total Volume = 2000 ft.*(0.0912 ft^2) = 182 $ft.^3$
Displacement Time = (182 $ft.^3$) / [50 spm*(1.1523 $ft.^3$/s)*60 min/hr] = 0.053 hrs.
Setup Time = 15 min / (60 min/hr) = 0.25 hrs.
Total Operation Time = 0.06 hrs. + 0.053 hrs. + 0.25 hrs. + 0.5 hrs. = 0.863 hrs.

Total Sacks Cement = 64 + 90 = 154 sacks
Total Time Required = 0.814 + 0.863 = 1.677 hours

**The pump used for this cement job is a duplex pump operating at 50 spm with a 90% volumetric efficiency. The pump has an 18 inch stroke, and consists of 2.5 inch rods and a 6.5 inch liner. The rate is equivalent to 1.1523 cubic feet per second.

Equations:

Critical Properties:

Tr = T/Tc

Pr = P/Pc

Ideal Gas Equation:

$$PV = nRT$$

Real Gas Equation:

$$PV = znRT$$

Redlich Kwong Equation of State:

$$P + \left(\frac{a}{\left(V(V + B) * \sqrt{T}\right)} \right)(V - b) = RT$$

Empirical Constants:

$$a = \Omega_a * \left(\frac{R_2 * T^{2.5}}{P_C} \right)$$

$$\Omega_a = .427480$$

$$b = \Omega_b * \left(\frac{R * T_C}{P_C} \right)$$

$$\Omega_b = .086640$$

Derived from the Redlich Kwong:

$$f(x_o) = \mathrm{z}^3 - \mathrm{z}^2 + \frac{\mathrm{P_r}}{\mathrm{T_r}}\left[\left(\frac{.42748}{\mathrm{T_r^{1.5}}}\right) - 0.08664 - .007506 * \frac{\mathrm{P_r}}{\mathrm{T_r}}\right] * z - .03704 * \frac{\mathrm{P_r^2}}{\mathrm{T_r^{3.5}}}$$

$$f'(x_o) = 3\mathrm{z}^2 - 2z + \frac{\mathrm{P_r}}{\mathrm{T_r}}\left[\left(\frac{.42748}{\mathrm{T_r^{1.5}}}\right) - 0.08664 - .007506 * \frac{\mathrm{P_r}}{\mathrm{T_r}}\right]$$

Newton Raphson Iteration Techniques:

$$x_1 = x_o - \left(\frac{f(x_o)}{f'(x_o)} \right)$$

Viscosity:

Viscosity = K*0.0001* exp(X*D*^Y)

Compressibility:

Cg = Cr / Pc

Porosity:

$\phi = 0.5(\phi D) + 0.5(\phi N)$

Initial Gas In Place:

IGIP = $[\phi * S_g]/B_{gi}$ (yields scf per acre-ft) $\rightarrow$ must be multiplied to the thickness

Formation Volume Factor:

$B_{gi} = 0.0283 * [z * T_i]/P_i$

Formation Temperature:

$T_f = T_s + (\text{Depth})T_g$

Flowing Wellhead Pressure:

$$P_{tf} = \left[\sqrt{P_{wf}{}^2 - \frac{25 g_g (T \bullet z \bullet f)_{avg} L(e^s - 1) q^2}{sD^5}} \right] / e^s$$

Deliverability Equation:

$$P_R^{\,2} - P_{wf}^{\,2} = aq + bq^2$$

where,

$$a = \frac{1422(mz)T}{kh}\left[\ln\left(\frac{0.472r_e}{r_w}\right) + S\right]$$

$$b = \frac{(3.161\times10^{-12})\,bzTg_g}{h^2 r_w}$$

$$b = \frac{2.7\times10^8}{k^{1.083}}$$

Pseudo-Pressure and Actual Time Approach for Build-Up Analysis:

$$k\ (\mathrm{mD}) = (1637*q*T)/(m*h)$$

$$S' = 1.151 * \{[(m(P_{wf}) - m(P_{ws})) / m] – [\log((k*\Delta t)/(\phi*\mu*c_t*(r_w^2)))] + 3.23\}$$

Pseudo-Pressure and Pseudo-Time Approach for Build-Up Analysis:

$$k = (1637*q*T)/(m*h)$$

$$S' = 1.151 * \{[(m(P_{wf}) - m(P_{ws})) / m] – [\log((k*\Delta t_{ap})/(\phi *(r_w^2)))] + 3.23 + [\log(t_{pap}+\Delta t_{ap}) / (t_{pap})] \}$$

$$C_{Deff} = \{[13.26*q*P_{sc}*T]/[\phi*h*(r_w^2)*T_{sc}]\} * [\Delta t_{ap}/\Delta m(P)]$$

Adjusted Pressure and Adjusted Time Approach for Build-Up Analysis:

$$k = -162.6 * [(q) * (B_{gavg}) * (\mu_{avg})] / (m * h)$$

$$S' = 1.151 * \{[(m(P_{wf}) - m(P_{ws})) / m] – [\log((k*\Delta t_A)/(\phi*\mu_{avg}*c_{tavg}*(r_w^2)))] + 3.23 + [\log(t_{pA}+\Delta t_A) / (t_{pA})] \}$$

Notation:

Critical Properties:

Tr = pseudo-reduced temperature
Pr = pseudo-reduced pressure
T = temperature
P = pressure
Tc = critical temperature
Pc = critical pressure

Ideal Gas Equation:

P = pressure
V = volume
N= moles
R = universal gas constant
T = temperature

Real Gas Equation:

P = pressure
V = volume
N= moles
R = universal gas constant
z = gas compressibility factor
T = temperature

Redlich Kwong Equation of State:

P = pressure
V = volume
N= moles
R = universal gas constant
Z = gas compressibility factor
T = temperature

Newton Raphson Iteration Techniques:

X_o = initial guess for z-factor

Viscosity:

K = ((9.4+0.02*m)*T^1.5)/(209+19*m + T)
Y = 2.4 – 0.2 * X
X = 3.5 + 986 / T + 0.01 * m
D = (m*P) / (Z *10.7 *T) * (1000/2.205) * (1/30.48^3)
M = Molecular Weight
T = Temperature, R
P = Pressure, psi
Z = Z-factor

Compressibility:

Cr = Reduced compressibility
Pc = critical pressure

Porosity:

ϕD = Density Porosity
ϕN = Neutron Porosity

Initial Gas In Place:

ϕ = porosity (fraction)
S_g = gas saturation (fraction)
B_{gi} = initial gas formation volume factor

Formation Volume Factor:

z = gas compressibility factor (z-factor)
T_i = initial reservoir temperature
P_i = initial reservoir pressure

Formation Temperature:

Depth = formation depth
T_s = mean annual surface temperature
T_g = temperature gradient (subsurface)

Flowing Wellhead Pressure:

P_{wf} = *flowing bottom-hole pressure (psi)*
P_{tf} = flowing wellhead pressure (psi)
s = $2g_g Z / 53.34 \times (Tz)_{avg}$
T_{avg} = avg. of bottomhole and wellhead pressure (Rankin)
z_{avg} = avg. compressibility factor at avg. temperature and pressure
f_{avg} = Moody friction factor at avg. temperature and pressure
L = length of flow string (feet)
Z = vertical depth of reservoir from surface (feet)
q = gas flow rate (MMcfd)
D = flow string diameter (inches)

Deliverability Equation:

μ = gas viscosity (cp)
z = gas compressibility factor
T = reservoir temperature (Rankin)
r_e = drainage radius (feet)
r_w = well radius (feet)
k = permeability (mD)
h = formation thickness (feet)
S = skin factor
a = deliverability coefficient
b = deliverability coefficient

Pseudo-Pressure and Actual Time Approach for Build-Up Analysis:

Permeability

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
m = slope of line (psi^2/cp/cycle)
h = formation thickness (feet)

Skin

$m(P_{wf})$ = flowing pressure of the well (psi^2/cp)
$m(P_{ws})$ = static pressure of the well (psi^2/cp)
m = slope of line (psi^2/cp/cycle)
ϕ = porosity (fraction)
μ = viscosity (cp)
c_t = compressibility (psi^{-1})
r_w = wellbore radius (feet)
Δt = SI time at $m(P_{ws})$
k = permeability (mD)

Pseudo-Pressure and Pseudo-Time Approach for Build-Up Analysis:

Permeability

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
m = slope of line (psi^2/cp/cycle)
h = formation thickness (feet)

Skin

$m(P_{wf})$ = flowing pressure of the well (psi^2/cp)
r_w = wellbore radius (feet)
$m(P_{ws})$ = static pressure of the well (psi^2/cp)
m = slope of line (psi^2/cp/cycle)
k = permeability (mD)
t_{pap} = Pseudo Production Time
Δt_{ap} = Pseudo-Time at $m(P_{ws})$
ϕ = porosity (fraction)

Wellbore Storage

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
T_{sc} = temperature at standard conditions (520 degrees Rankin)
h = formation thickness (feet)
r_w = wellbore radius (feet)
P_{sc} = pressure at standard conditions (14.7 psia)
ϕ = porosity (fraction)
Δt_{ap} = Pseudo-Time at $\Delta m(P)$
$\Delta m(P)$ = difference in Pseudo-Pressure at Δt_{ap}

Adjusted Pressure and Adjusted Time Approach for Build-Up Analysis:

Permeability

q = gas flow rate (Mcfd)
T = reservoir temperature (Rankin)
m = slope of line (psi^2/cp/cycle)
h = formation thickness (feet)
μ_{avg} = viscosity at average pressure (cp)
B_{gavg} = average gas formation volume factor (RB/Mscf)

Skin

$m(P_{wf})$ = flowing pressure of the well (psi^2/cp)
$m(P_{ws})$ = static pressure of the well (psi^2/cp)
m = slope of line (psi^2/cp/cycle)
k = permeability (mD)
t_{pA} = Pseudo Production Time
Δt_A = Pseudo-Time at $m(P_{ws})$
ϕ = porosity (fraction)
μ_{avg} = viscosity at average pressure (cp)
r_w = wellbore radius (feet)
c_{tavg} = compressibility at average pressure (psi^{-1})

Source Code:

Flow Optimization Program:

Module:

```
'All variables must be declared'
Option Explicit
Public i, j As Integer
Public Const PI = 3.141592654

Public P, P1, P2, P3 As Single
Public degrees_F, T, zn1 As Single
Public Ma, Ppc, Tpc As Single
Public Impurities, corr, Tpccorr, Ppccorr, Tpr, Ppr, Z, Cpr, mhu As Single
Public A, B, C, D, E, F, G, H, I2, SG, SG7plus As Single
Public C1, C2, C3, IC4, NC4, IC5, NC5, C6, C7, C7plus, N2, H2S, CO2 As Single
Public M1, M2, M3, MI4, MN4, MI5, MN5, M6, M7, M7p, M7plus, MN2, MH2S, MCO2 As Single
Public Tc1, Tc2, Tc3, TcI4, TcN4, TcI5, TcN5, Tc6, Tc7, Tc7p, Tc7plus, TcN2, TcH2S, TcCO2 As Single
Public Pc1, Pc2, Pc3, PcI4, PcN4, PcI5, PcN5, Pc6, Pc7, Pc7p, Pc7plus, PcN2, PcH2S, PcCO2 As Single
Public rho, dzdrho As Single
Public A1, A2, A3, A4, A5, A6, A7, A8 As Single
Public rhog, K, X, Y, Visc, Cg As Single
Public NRe, IGIP, friction, Bg, Bgi, aa, beta, bb, qmax, re, rw, skin As Double
Public Pabandon, Time, YrTime, T1, T2, T3, D1, D2, D3, h1, h2, h3, por1, por2, por3 As Double
Public q1, q2, q3, Tp1, Tp2, Tp3, SG1, SG2, SG3, rw1, rw2, rw3, Depth1, Depth2, Depth3 As Double
Public Perm1, Perm2, Perm3, skin1, skin2, skin3, SP1, SP2, SP3, IGIP1, IGIP2, IGIP3 As Double
Public ResA1, ResA2, ResA3, qinit, qfin, Fo, F1, F2, z1 As Double

Public Pressure1(1100) As Double
Public Pressure2(1100) As Double
Public Pressure3(1100) As Double
Public Gptot(1100) As Double
Public Pwf(1100) As Double
Public Ptf(1100) As Double
Public Viscfin(1100) As Double
Public YrTimetot(1100) As Double
```

Input Form:

```
Private Sub cmdExit_Click()
End
End Sub

Public Sub Gas_properties()
   Ppc = 709.604 - (58.718 * SG)
   Tpc = 170.491 + (307.344 * SG)

   Ppr = P / Ppc
   Tpr = T / Tpc
End Sub


Private Sub Command1_Click()
PB1.Visible = True:        PB1.Min = 1:        PB1.Max = 10:

P1 = Val(txtpress1.Text)
T1 = ((Val(txttemp1.Text) + 460#) + (525#)) / 2
D1 = Val(txtd1.Text)
h1 = Val(txth1.Text)
```

```
por1 = (Val(txtpor1.Text)) / 100#
SG1 = Val(txtsg1.Text)
rw1 = Val(txtrw1.Text)
Depth1 = Val(txtDepth1.Text)
Perm1 = Val(txtPerm1.Text)
skin1 = Val(txtSkin1.Text)
SP1 = Val(txtSP1.Text)
IGIP1 = Val(txtIGIP1.Text)
ResA1 = Val(txtResA1.Text)

PB1.Value = 1

'Initial Gas in Place'
IGIPfin1 = IGIP1 * ResA1

'Drainage Area'
re1 = (ResA1 / PI) ^ 0.5

'Increment q and pressure'
        P = P1
        T = T1
        SG = SG1
        Gas_properties
        Z_factor
        Cg_Compressibility
        Visc_Viscosity

Bgi = (0.00504 * Z * (T1)) / P

aa = (((1422) * (Visc) * (Z) * (T1)) / ((Perm1) * (h1))) * (Log((0.472) * (re1 / rw1)) + skin1)
aa = Abs(aa)
beta = (270000000) / ((Perm1) ^ 1.083)
bb = ((0.000000000003161) * (beta) * (Z) * (T1) * (SG1)) / ((h1) ^ 2 * (rw1))

First = (((aa) ^ 2#) + (4#) * (bb) * ((P) ^ 2)) ^ 0.5
qmax = (((-aa) + First) / ((2#) * (bb)))

PB1.Value = 2

    For j = 0 To 1000
    Pressure1(j) = P1 - ((j * P1) / 1000)
    Next j

PB1.Value = 3

YrTime = 0

'Calculate Gas Properties, Pwf, Ptf by Incrementing Pressure'
For j = 1 To (qmax + 1000) Step 1
 If YrTime > 7 And YrTime < 8 Then
    GoTo 3
 Else
    qinit = j * 0.01
    q = qinit
3:
    qfin = q + 0.001
    q = qfin
End If
```

```
   For i = 0 To 1000
      P = Pressure1(i)
      Gas_properties
      Z_factor
      Cg_Compressibility
      Visc_Viscosity

Viscfin(i) = Visc

      aa = (((1422) * (Visc) * (Z) * (T1)) / ((Perm1) * (h1))) * (Log((0.427) * (re1 / rw1)) + skin1)
      beta = (270000000) / ((Perm1) ^ 1.083)
      bb = ((0.000000000003161) * (beta) * (Z) * (T1) * (SG1)) / ((h1) ^ 2 * (rw1))

If (((P) ^ 2) - ((aa) * (q)) - ((bb) * ((q) ^ 2))) < 0 Then
   GoTo 6
Else
      Pwf(i) = (((P) ^ 2) - ((aa) * (q)) - ((bb) * ((q) ^ 2))) ^ 0.5
End If

   NRe = (20 * (q) * (SG1)) / ((Visc) * (D1))
   If NRe = 0 Then GoTo 1
friction = 1 / (1.14 - (2# * Log((0.0006 / (D1)) + (21.25 / ((NRe) ^ 0.9)))))
1:

   vars = (2# * (SG1) * (Depth1)) / (53.34 * (T1) * (Z))
   evars = Exp(vars)

   one = ((Pwf(i)) ^ 2#) / (evars)
   two = (25# * (SG1) * (T) * (Z) * (friction) * (Depth1) * (evars - 1#) * (((q) / 1000) ^ 2#))
   three = vars * ((D1) ^ 5#)
   Ptf(i) = (one - (((two) / (three)) / evars)) ^ 0.5

   If P = 0 Then
   Gptot(i) = 0
   GoTo 5
   End If

'Calculate Gp and Time'
   Bg = (0.00504 * Z * (T1)) / P
   Gptot(i) = (IGIP1 * (Bg - Bgi)) / (Bg)
5:

If q = 0 Then
Time = 0
YrTime = 0
Else
   Time = (Gptot(i) / (1000 * q))
   YrTime = Time / 365
   YrTimetot(i) = YrTime
End If
   Pabandon = SP1 + 50

'Checking pressure'
If Ptf(i) < Pabandon Then
   GoTo 8
End If
```

```
Next i
8:

Counter = i
frm295.MSWeir.Rows = (Counter + 2)

'Checking Time'
If YrTime < 7.01 And YrTime > 6.99 Then
GoTo 9
End If
6:
Next j
9:

PB1.Value = 8

'Display Results'
Open "a:weir.txt" For Output As #1
Print #1, "Pressure", "Flow Rate", "Pwf", "WH Pressure", "Viscosity", "Time (yr)"
For i = 0 To Counter
Print #1, Format(Val(Pressure1(i)), "0.00"), q, Format(Val(Pwf(i)), "0.00"), Format(Val(Ptf(i)), "0.00"),
Format(Val(Viscfin(i)), "0.000000"), Format(Val(YrTimetot(i)), "0.00")
Next i
Close #1

For i = 0 To Counter
frm295.MSWeir.Col = 0
frm295.MSWeir.Row = i + 1
frm295.MSWeir.CellAlignment = 3
frm295.MSWeir.Text = Format(Val(Pressure1(i)), "0.00")

frm295.MSWeir.Col = 1
frm295.MSWeir.Row = i + 1
frm295.MSWeir.CellAlignment = 3
frm295.MSWeir.Text = Format(Val(q), "0.00")

frm295.MSWeir.Col = 2
frm295.MSWeir.Row = i + 1
frm295.MSWeir.CellAlignment = 3
frm295.MSWeir.Text = Format(Val(Pwf(i)), "0.00")

frm295.MSWeir.Col = 3
frm295.MSWeir.Row = i + 1
frm295.MSWeir.CellAlignment = 3
frm295.MSWeir.Text = Format(Val(Ptf(i)), "0.00")

frm295.MSWeir.Col = 4
frm295.MSWeir.Row = i + 1
frm295.MSWeir.CellAlignment = 3
frm295.MSWeir.Text = Format(Val(Viscfin(i)), "0.000000")

frm295.MSWeir.Col = 5
frm295.MSWeir.Row = i + 1
frm295.MSWeir.CellAlignment = 3
frm295.MSWeir.Text = Format(Val(YrTimetot(i)), "0.00")
Next i

frm295.txtday.Text = Format(Val(Time), "0.00")
```

```
frm295.txtyear.Text = Format(Val(YrTime), "0.00")

PB1.Value = 10

End Sub

Public Sub Z_factor()
'Z-Factor'
'   If Ppr > 0.2 And Ppr < 1.2 Then
'      If Tpr > 1.05 And Tpr < 1.2 Then
'         Z = Ppr * ((1.6643 * Tpr) - 2.2114) - (0.3647 * Tpr) + 1.4385
'      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
'         Z = Ppr * ((0.522 * Tpr) - 0.8511) - (0.0364 * Tpr) + 1.049
'      ElseIf Tpr > 1.4 And Tpr < 2# Then
'         Z = Ppr * ((0.1391 * Tpr) - 0.2988) + (0.0007 * Tpr) + 0.9969
'      ElseIf Tpr > 2# And Tpr < 3# Then
'         Z = Ppr * ((0.0295 * Tpr) - 0.0825) + (0.0009 * Tpr) + 0.9967
'      End If
'   '-------------------------------------------------
'   ElseIf Ppr > 1.2 And Ppr < 2.8 Then
'      If Tpr > 1.05 And Tpr < 1.2 Then
'         Z = Ppr * ((-1.357 * Tpr) + 1.4942) + (4.6315 * Tpr) - 4.7009
'      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
'         Z = Ppr * ((0.1717 * Tpr) - 0.3232) + (0.5869 * Tpr) + 0.1229
'      ElseIf Tpr > 1.4 And Tpr < 2# Then
'         Z = Ppr * ((0.0984 * Tpr) - 0.2053) + (0.0621 * Tpr) + 0.858
'      ElseIf Tpr > 2# And Tpr < 3# Then
'         Z = Ppr * ((0.0211 * Tpr) - 0.0527) + (0.0127 * Tpr) + 0.9549
'      End If
'   '-------------------------------------------------
'   ElseIf Ppr > 2.8 And Ppr < 5.4 Then
'      If Tpr > 1.05 And Tpr < 1.2 Then
'         Z = Ppr * ((-0.3278 * Tpr) + 0.4752) + (1.8223 * Tpr) - 1.9036
'      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
'         Z = Ppr * ((-0.2521 * Tpr) + 0.3871) + (1.6087 * Tpr) - 1.6635
'      ElseIf Tpr > 1.4 And Tpr < 2# Then
'         Z = Ppr * ((-0.0284 * Tpr) + 0.0625) + (0.4714 * Tpr) - 0.0011
'      ElseIf Tpr > 2# And Tpr < 3# Then
'         Z = Ppr * ((0.0041 * Tpr) + 0.0039) + (0.0607 * Tpr) + 0.7927
'      End If
'   '-------------------------------------------------
'   ElseIf Ppr > 5.4 And Ppr < 15# Then
'      If Tpr > 1.05 And Tpr < 3# Then
'         Z = Ppr * ((0.711 + (3.66 * Tpr)) ^ (-1.4667)) - (1.637 / ((0.319 * Tpr) + 0.522)) + 2.071
'      End If
'   '-------------------------------------------------
'   ElseIf Ppr > 15# Then
'         Z = 1
'   End If
'   '-------------------------------------------------
'   If Z < 0.25 Then
'      Z = 10 * Z
'   End If

   Z = 1
   Do
F1 = (Ppr / Tpr) * ((0.4278 / (Tpr ^ 1.5)) - 0.08664 - (0.007506 * (Ppr / Tpr))) * Z - ((0.03704) * ((Ppr ^ 2)
/ (Tpr ^ 3.5)))
```

```
    Fo = (Z ^ 3) - (Z ^ 2) + F1
    F2 = (3 * (Z ^ 2)) - (2 * Z) + (Ppr / Tpr) * ((0.42748 / (Tpr ^ 1.5)) - 0.08664 - (0.007506 * (Ppr / Tpr)))
    z1 = (Z - (Fo / F2))

    If Abs(z1 - Z) < 0.01 Then
    Z = z1
    GoTo 10
    Else
    Z = z1
    End If
    Loop
10
End Sub

Public Sub Cg_Compressibility()
 A1 = 0.31506237:    A2 = -1.0467099:    A3 = 0.057832729:    A4 = 0.53530771
    A5 = -0.61232032:    A6 = -0.10488813:    A7 = 0.68157001:    A8 = 0.68446549

    rho = 0.27 * (Ppr / (Z * Tpr))
    D = A1 + (A2 / Tpr) + (A3 / Tpr ^ 3)
    E = 2 * (A4 + (A5 / Tpr)) * rho
    F = 5 * (A5) * (A6) * (rho ^ 4) / Tpr
    G = 2 * (A7) * (rho ^ 3)
    H = 1 + (A8 * (rho ^ 2)) - ((A8 ^ 2) * (rho ^ 4))
    I2 = Exp(-A8 * (rho ^ 2))

    dzdrho = D + E + F + (G * H * i)
    Cpr = (1 / Ppr) - ((0.27 / (Z ^ 2 * Tpr)) * ((dzdrho * Tpr) / (1 + (rho / Z) * dzdrho)))
    Cg = Cpr / Ppc

End Sub

Public Sub Visc_Viscosity()
'Viscosity (SG) '
    Ma = SG * 29#

    rhog = ((2.7 * (SG) * (P)) / ((Z) * (T))) * 0.016018
    K = ((9.4 + (0.02 * Ma)) * (T ^ 1.5)) / (209 + (19 * Ma) + T)
    X = 3.5 + (986 / T) + (0.01) * (Ma)
    Y = 2.4 - (0.2 * X)
    Visc = (K * 10 ^ (-4)) * (Exp(X * (rhog ^ Y)))

End Sub

Private Sub Command2_Click()
PB2.Visible = True:         PB2.Min = 1:         PB2.Max = 10:

P2 = Val(txtpress2.Text)
T2 = ((Val(txttemp2.Text) + 460#) + (525#)) / 2
D2 = Val(txtd2.Text)
h2 = Val(txth2.Text)
por2 = (Val(txtpor2.Text)) / 100#
SG2 = Val(txtsg2.Text)
rw2 = Val(txtrw2.Text)
Depth2 = Val(txtDepth2.Text)
Perm2 = Val(txtPerm2.Text)
skin2 = Val(txtSkin2.Text)
SP2 = Val(txtSP2.Text)
```

```
IGIP2 = Val(txtIGIP2.Text)
ResA2 = Val(txtResA2.Text)

PB2.Value = 1

IGIPfin2 = IGIP2 * ResA2
re2 = (ResA2 / PI) ^ 0.5

    P = P2
    T = T2
    SG = SG2
    Gas_properties
    Z_factor
    Cg_Compressibility
    Visc_Viscosity

Bgi = (0.00504 * Z * (T2)) / P

aa = (((1422) * (Visc) * (Z) * (T2)) / ((Perm2) * (h2))) * (Log((0.472) * (re2 / rw2)) + skin2)
aa = Abs(aa)
beta = (270000000) / ((Perm2) ^ 1.083)
bb = ((0.000000000003161) * (beta) * (Z) * (T2) * (SG2)) / ((h2) ^ 2 * (rw2))

First = (((aa) ^ 2#) + (4#) * (bb) * ((P) ^ 2)) ^ 0.5
qmax = ((-aa) + First) / ((2#) * (bb))

PB2.Value = 2

  For j = 0 To 1000
  Pressure2(j) = P2 - ((j * P2) / 1000)
  Next j

PB2.Value = 3

For j = 1 To (qmax + 1000) Step 1
 If YrTime > 7 And YrTime < 8 Then
  GoTo 3
 Else
  qinit = j * 0.01
q = qinit
3:
  qfin = q + 0.001
  q = qfin
End If

  For i = 0 To 1000
    P = Pressure2(i)
    Gas_properties
    Z_factor
    Cg_Compressibility
    Visc_Viscosity

Viscfin(i) = Visc

    aa = (((1422) * (Visc) * (Z) * (T2)) / ((Perm2) * (h2))) * (Log((0.427) * (re2 / rw2)) + skin2)
    aa = Abs(aa)
    beta = (270000000) / ((Perm2) ^ 1.083)
    bb = ((0.000000000003161) * (beta) * (Z) * (T2) * (SG2)) / ((h2) ^ 2 * (rw2))
```

```
If (((P) ^ 2) - ((aa) * (q)) - ((bb) * ((q) ^ 2))) < 0 Then
   GoTo 6
Else
      Pwf(i) = (((P) ^ 2) - ((aa) * (q)) - ((bb) * ((q) ^ 2))) ^ 0.5
End If

   NRe = (20 * (q) * (SG2)) / ((Visc) * (D2))
   If NRe = 0 Then GoTo 1
      friction = 1 / (1.14 - (2# * Log((0.0006 / (D2)) + (21.25 / ((NRe) ^ 0.9)))))
1:

   vars = (2# * (SG2) * (Depth2)) / (53.34 * (T2) * (Z))
   evars = Exp(vars)

   one = ((Pwf(i)) ^ 2#) / (evars)
   two = (25# * (SG2) * (T2) * (Z) * (friction) * (Depth2) * (evars - 1#) * (((q) / 1000) ^ 2#))
   three = vars * ((D2) ^ 5#)
   Ptf(i) = (one - (((two) / (three)) / evars)) ^ 0.5

   If P = 0 Then
   Gptot(i) = 0
   GoTo 5
   End If

   Bg = (0.00504 * Z * (T2)) / P
   Gptot(i) = (IGIP2 * (Bg - Bgi)) / (Bg)
5:

If q = 0 Then
Time = 0
YrTime = 0
Else
   Time = (Gptot(i) / (1000 * q))
   YrTime = Time / 365
   YrTimetot(i) = YrTime
End If

   Pabandon = SP2 + 50#

If Ptf(i) < Pabandon Then
   GoTo 8
End If
Next i
8:

Counter = i
frmstrange.MSBlue.Rows = (Counter + 2)

If YrTime < 7.01 And YrTime > 6.99 Then
GoTo 9
End If
6:
Next j
9:

PB2.Value = 8
```

```
Open "a:bluem.txt" For Output As #1
Print #1, "Pressure", "Flow Rate", "Pwf", "WH Pressure", "Viscosity", "Time (yr)"
For i = 0 To Counter
Print #1, Format(Val(Pressure2(i)), "0.00"), Format(Val(q), "0.00"), Format(Val(Pwf(i)), "0.00"),
Format(Val(Ptf(i)), "0.00"), Format(Val(Viscfin(i)), "0.00000"), Format(Val(YrTimetot(i)), "0.00")
Next i
Close #1

For i = 0 To Counter
frmstrange.MSBlue.Col = 0
frmstrange.MSBlue.Row = i + 1
frmstrange.MSBlue.CellAlignment = 3
frmstrange.MSBlue.Text = Format(Val(Pressure2(i)), "0.00")

frmstrange.MSBlue.Col = 1
frmstrange.MSBlue.Row = i + 1
frmstrange.MSBlue.CellAlignment = 3
frmstrange.MSBlue.Text = Format(Val(q), "0.00")

frmstrange.MSBlue.Col = 2
frmstrange.MSBlue.Row = i + 1
frmstrange.MSBlue.CellAlignment = 3
frmstrange.MSBlue.Text = Format(Val(Pwf(i)), "0.00")

frmstrange.MSBlue.Col = 3
frmstrange.MSBlue.Row = i + 1
frmstrange.MSBlue.CellAlignment = 3
frmstrange.MSBlue.Text = Format(Val(Ptf(i)), "0.00")

frmstrange.MSBlue.Col = 4
frmstrange.MSBlue.Row = i + 1
frmstrange.MSBlue.CellAlignment = 3
frmstrange.MSBlue.Text = Format(Val(Viscfin(i)), "0.00000")

frmstrange.MSBlue.Col = 5
frmstrange.MSBlue.Row = i + 1
frmstrange.MSBlue.CellAlignment = 3
frmstrange.MSBlue.Text = Format(Val(YrTimetot(i)), "0.00")
Next i

frmstrange.txtday2.Text = Format(Val(Time), "0.00")
frmstrange.txtyear2.Text = Format(Val(YrTime), "0.00")

PB2.Value = 10

End Sub

Private Sub Command3_Click()
PB3.Visible = True:     PB3.Min = 1:     PB3.Max = 10:

P3 = Val(txtpress3.Text)
T3 = ((Val(txttemp3.Text) + 460#) + (525#)) / 2
D3 = Val(txtd3.Text)
h3 = Val(txth3.Text)
por3 = (Val(txtpor3.Text)) / 100#
SG3 = Val(txtsg3.Text)
rw3 = Val(txtrw3.Text)
Depth3 = Val(txtDepth3.Text)
```

```
Perm3 = Val(txtPerm3.Text)
skin3 = Val(txtSkin3.Text)
SP3 = Val(txtSP3.Text)
IGIP3 = Val(txtIGIP3.Text)
ResA3 = Val(txtResA3.Text)

PB3.Value = 1

IGIPfin3 = IGIP3 * ResA3
re3 = (ResA3 / PI) ^ 0.5

    P = P3
    T = T3
    SG = SG3
    Gas_properties
Z_factor
    Cg_Compressibility
    Visc_Viscosity

Bgi = (0.00504 * Z * (T3)) / P

aa = (((1422) * (Visc) * (Z) * (T3)) / ((Perm3) * (h3))) * (Log((0.472) * (re3 / rw3)) + skin3)
aa = Abs(aa)
beta = (270000000) / ((Perm3) ^ 1.083)
bb = ((0.000000000003161) * (beta) * (Z) * (T3) * (SG3)) / ((h3) ^ 2 * (rw3))

First = (((aa) ^ 2#) + (4#) * (bb) * ((P) ^ 2)) ^ 0.5
qmax = ((-aa) + First) / ((2#) * (bb))

PB3.Value = 2

  For j = 0 To 1000
  Pressure3(j) = P3 - ((j * P3) / 1000)
  Next j

PB3.Value = 3

For j = 1 To (qmax + 1000) Step 1
 If YrTime > 7 And YrTime < 8 Then
  GoTo 3
 Else
  qinit = j * 0.01
  q = qinit
3:
  qfin = q + 0.001
  q = qfin
End If

  For i = 0 To 1000
    P = Pressure3(i)
    Gas_properties
    Z_factor
    Cg_Compressibility
    Visc_Viscosity

Viscfin(i) = Visc

    aa = (((1422) * (Visc) * (Z) * (T3)) / ((Perm3) * (h3))) * (Log((0.472) * (re3 / rw3)) + skin3)
```

```
        beta = (270000000) / ((Perm3) ^ 1.083)
        bb = ((0.000000000003161) * (beta) * (Z) * (T3) * (SG3)) / ((h3) ^ 2 * (rw3))

If (((P) ^ 2) - ((aa) * (q)) - ((bb) * ((q) ^ 2))) < 0 Then
    GoTo 6
Else
        Pwf(i) = (((P) ^ 2) - ((aa) * (q)) - ((bb) * ((q) ^ 2))) ^ 0.5
End If

    NRe = (20 * (q) * (SG3)) / ((Visc) * (D3))
    If NRe = 0 Then GoTo 1
        friction = 1 / (1.14 - (2# * Log((0.0006 / (D3)) + (21.25 / ((NRe) ^ 0.9)))))
1:

    vars = (2# * (SG3) * (Depth3)) / (53.34 * (T3) * (Z))
    evars = Exp(vars)

    one = ((Pwf(i)) ^ 2#) / (evars)
    two = (25# * (SG3) * (T3) * (Z) * (friction) * (Depth3) * (evars - 1#) * (((q) / 1000) ^ 2#))
    three = vars * ((D3) ^ 5#)
    Ptf(i) = (one - (((two) / (three)) / evars)) ^ 0.5

    If P = 0 Then
    Gptot(i) = 0
    GoTo 5
    End If

    Bg = (0.00504 * Z * (T)) / P
    Gptot(i) = (IGIP3 * (Bg - Bgi)) / (Bg)
5:

If q = 0 Then
Time = 0
YrTime = 0
Else
    Time = (Gptot(i) / (1000 * q))
    YrTime = Time / 365
    YrTimetot(i) = YrTime
End If
    Pabandon = SP3 + 50

If Ptf(i) < Pabandon Then
    GoTo 8
End If
Next i
8:

Counter = i
frmstrange.MSBig.Rows = (Counter + 2)

If YrTime < 7.01 And YrTime > 6.99 Then
GoTo 9
End If
6:
Next j
9:

PB3.Value = 8
```

```
Open "a:biginj.txt" For Output As #1
Print #1, "Pressure", "Flow Rate", "Pwf", "WH Pressure", "Viscosity", "Time (yr)"
For i = 0 To Counter
Print #1, Format(Val(Pressure3(i)), "0.00"), Format(Val(q), "0.00"), Format(Val(Pwf(i)), "0.00"),
Format(Val(Ptf(i)), "0.00"), Format(Val(Viscfin(i)), "0.00000"), Format(Val(YrTimetot(i)), "0.00")
Next i
Close #1

For i = 0 To Counter
frmstrange.MSBig.Col = 0
frmstrange.MSBig.Row = i + 1
frmstrange.MSBig.CellAlignment = 3
frmstrange.MSBig.Text = Format(Val(Pressure3(i)), "0.00")

frmstrange.MSBig.Col = 1
frmstrange.MSBig.Row = i + 1
frmstrange.MSBig.CellAlignment = 3
frmstrange.MSBig.Text = Format(Val(q), "0.00")

frmstrange.MSBig.Col = 2
frmstrange.MSBig.Row = i + 1
frmstrange.MSBig.CellAlignment = 3
frmstrange.MSBig.Text = Format(Val(Pwf(i)), "0.00")

frmstrange.MSBig.Col = 3
frmstrange.MSBig.Row = i + 1
frmstrange.MSBig.CellAlignment = 3
frmstrange.MSBig.Text = Format(Val(Ptf(i)), "0.00")

frmstrange.MSBig.Col = 4
frmstrange.MSBig.Row = i + 1
frmstrange.MSBig.CellAlignment = 3
frmstrange.MSBig.Text = Format(Val(Viscfin(i)), "0.00000")

frmstrange.MSBig.Col = 5
frmstrange.MSBig.Row = i + 1
frmstrange.MSBig.CellAlignment = 3
frmstrange.MSBig.Text = Format(Val(YrTimetot(i)), "0.00")
Next i

frmstrange.txtday3.Text = Format(Val(Time), "0.00")
frmstrange.txtyear3.Text = Format(Val(YrTime), "0.00")

PB3.Value = 10

End Sub

Private Sub Command4_Click()
frm295.Show
FrmPnge270.Hide
End Sub

Private Sub Command5_Click()
frmstrange.Show
FrmPnge270.Hide

End Sub
```

```
Private Sub Text1_Change()

End Sub
```

Results Forms (Elkhurst Well):

```
Private Sub cmdBack_Click()
frm295.Hide
FrmPnge270.Show

End Sub

Private Sub cmdExit_Click()
End

End Sub

Private Sub Command1_Click()
End
End Sub

Private Sub Command2_Click()
FrmPnge270.Show
frm295.Hide

End Sub

Private Sub Text1_Change()

End Sub

Private Sub Command3_Click()

End Sub

Private Sub Form_Load()
Dim i1 As Integer

For i1 = 1 To MSWeir.Cols - 1
MSWeir.ColWidth(i1) = 2000
Next i1

MSWeir.Row = 0

MSWeir.Col = 0
frm295.MSWeir.CellAlignment = 3
MSWeir.Text = "Pressure (psi)"

MSWeir.Col = 1
frm295.MSWeir.CellAlignment = 3
MSWeir.Text = "Flow Rate (Mcfd)"

MSWeir.Col = 2
frm295.MSWeir.CellAlignment = 3
MSWeir.Text = "Pwf (psi)"

MSWeir.Col = 3
frm295.MSWeir.CellAlignment = 3
```

```
MSWeir.Text = "WH Pressure (psi)"

MSWeir.Col = 4
frm295.MSWeir.CellAlignment = 3
MSWeir.Text = "Viscosity (cp)"

MSWeir.Col = 5
frm295.MSWeir.CellAlignment = 3
MSWeir.Text = "Time (hr)"
End Sub
```

Results Forms (Strange Creek Well):

```
Option Explicit

Private Sub Command1_Click()
FrmPnge270.Show
frmstrange.Hide

End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Form_Load()
Dim i1 As Integer

For i1 = 1 To MSBlue.Cols - 1
MSBlue.ColWidth(i1) = 2000
Next i1

For i1 = 1 To MSBig.Cols - 1
MSBig.ColWidth(i1) = 2000
Next i1

MSBlue.Row = 0
MSBig.Row = 0

MSBlue.Col = 0
MSBlue.CellAlignment = 3
MSBlue.Text = "Pressure (psi)"

MSBlue.Col = 1
MSBlue.CellAlignment = 3
MSBlue.Text = "Flow Rate (Mcfd)"

MSBlue.Col = 2
MSBlue.CellAlignment = 3
MSBlue.Text = "Pwf (psi)"

MSBlue.Col = 3
MSBlue.CellAlignment = 3
MSBlue.Text = "WH Pressure (psi)"

MSBlue.Col = 4
MSBlue.CellAlignment = 3
MSBlue.Text = "Viscosity (cp)"
```

```
MSBlue.Col = 5
MSBlue.CellAlignment = 3
MSBlue.Text = "Time (yr)"

MSBig.Col = 0
MSBig.CellAlignment = 3
MSBig.Text = "Pressure (psi)"

MSBig.Col = 1
MSBig.CellAlignment = 3
MSBig.Text = "Flow Rate (Mcfd)"

MSBig.Col = 2
MSBig.CellAlignment = 3
MSBig.Text = "Pwf (psi)"

MSBig.Col = 3
MSBig.CellAlignment = 3
MSBig.Text = "WH Pressure (psi)"

MSBig.Col = 4
MSBig.CellAlignment = 3
MSBig.Text = "Viscosity (cp)"

MSBig.Col = 5
MSBig.CellAlignment = 3
MSBig.Text = "Time (yr)"

End Sub
```

Standing-Katz Program:

Module:

```
Option Explicit

Public Tp(20) As Double
Public Add, z As Double
```

Form:

```
Private Sub Command1_Click()
Dim nameTpr As String
Dim pathnameTpr As String

pathnameTpr = drvTpr.Drive
nameTpr = pathnameTpr + "\" + txtTpr.Text + ".txt"
Open nameTpr For Input As #1
   i = 1
   Add = 1
Do While Not EOF(1)
Input #1, Tp(i)

   i = i + 1
   Add = Add + 1

Loop
Close #1
End Sub

Private Sub Command2_Click()
ReDim Alan(0 To 151, 0 To 151)
P = Val(Text1.Text)
SG = Val(Text2.Text)
Ppc = 709.604 - (58.718 * SG)
Pprm = 15
inc = Pprm / 150
i = 0
For i = 1 To Add - 1
   Ppr = 0
   Tpr = Tp(i)
   If Tpr = 0 Then
   GoTo 1
   End If
   For j = 1 To 150
         Ppr = Ppr + inc
         Alan(j, 0) = Ppr
     Call Z_FACT(Ppr, Tpr)
      Alan(j, i) = z
    Next j
1:
Next i

'Open "A:RESULTS.TXT" For Output As #2
 '   For j = 1 To 150
'  Print #2, Alan(j, 2), Alan(j, 3), Alan(j, 4), Alan(j, 5), Alan(j, 6), Alan(j, 7), Alan(j, 8), Alan(j, 9), Alan(j,
10)
 '  Next
With MSChart1
   .ColumnCount = 18
```

```
   .RowCount = 150

For k1 = 1 To Add - 1
   For k = 1 To 2
     No_Columns = No_Columns + 1
      For i = 1 To 150 Step 1
         .Column = No_Columns
         .Row = i
         If k = 1 Then
         .Data = Alan(i, 0)
         Else
         .Data = Alan(i, k1)
         End If
      Next
   Next
Next
End With
MSChart1.Plot.UniformAxis = False
End Sub
Function Z_FACT(Ppr, Tpr)
   z = 1
   Do
   F1 = (Ppr / Tpr) * ((0.4278 / (Tpr ^ 1.5)) - 0.08664 - (0.007506 * (Ppr / Tpr))) * z - ((0.03704) * ((Ppr ^
2) / (Tpr ^ 3.5)))
   Fo = (z ^ 3) - (z ^ 2) + F1
   F2 = (3 * (z ^ 2)) - (2 * z) + (Ppr / Tpr) * ((0.42748 / (Tpr ^ 1.5)) - 0.08664 - (0.007506 * (Ppr / Tpr)))
   z1 = (z - (Fo / F2))

   If Abs(z1 - z) < 0.01 Then
   z = z1
   GoTo 10
   Else
   z = z1
   End If
   Loop
10
End Function
Private Sub Command3_Click()
End
End Sub
```

Well Test Analysis Program:

Intro:

```
Private Sub Command1_Click()
frmIntro2.Hide
frmzfactor2.Show
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Form_Load()

End Sub
```

z-factor:

```
Private Sub cmdContinue_Click()

End Sub

Private Sub cmdex10_Click()
optGasComposition.Value = True
txtC1.Text = 78#
txtC2.Text = 0.43
txtC3.Text = 0.07
txtIC4.Text = 0.05
txtNC4.Text = 0.03
txtIC5.Text = 0.01
txtNC5.Text = 0.01
txtC6.Text = 0.01
txtC7.Text = 0#
txtC7plus.Text = 0#
txtC8.Text = 0.04
txtC9.Text = 0#
txtC10.Text = 0#
txtH2S.Text = 18.41
txtN2.Text = 1.3
txtCO2.Text = 1.64
txtTemp.Text = 180
End Sub

Private Sub cmdex9_Click()
optGasComposition.Value = True
txtC1.Text = 84.7
txtC2.Text = 5.86
txtC3.Text = 2.2
txtIC4.Text = 0.35
txtNC4.Text = 0.58
txtIC5.Text = 0.27
txtNC5.Text = 0.25
txtC6.Text = 0.28
txtC7.Text = 0.28
txtC7plus.Text = 0#
txtC8.Text = 0.15
```

```
txtC9.Text = 0.18
txtC10.Text = 0.15
txtH2S.Text = 0#
txtN2.Text = 3.45
txtCO2.Text = 1.3
txtTemp.Text = 150
End Sub

Private Sub cmdExit_Click()
End
End Sub

Private Sub Text1_Change()

End Sub

Private Sub lblZFactor_Click()
End Sub

Public Sub txtPressure_Change()
End Sub

Private Sub cmdImport_Click()
Dim namePressSITime As String
Dim pathnamePressSITime As String
Pinitial = Val(txtPinitial.Text)
PT = Val(txtProdTime.Text)

pathnamePressSITime = drvPressSITime.Drive
namePressSITime = pathnamePressSITime + "\" + txtPressSITime + ".txt"
Open namePressSITime For Input As #1
   i = 1
Do While Not EOF(1)
Input #1, P(i), SITime(i)
   i = i + 1
Loop
Close #1
counter = i
End Sub

Private Sub cmdPTAdj_Click()

End Sub

Private Sub cmdPT_Click()
Pinitial = Val(txtPinitial.Text)
PT = Val(txtProdTime.Text)

degrees_F = Val(txtTemp.Text)
T = degrees_F + 460#

P(0) = 100000
For j = 1 To counter - 1
   If P(j) < P(j - 1) Then
   minP = P(j)
   End If
Next j
minP = minP
```

```
PAverage = Int((Pinitial + minP) / 2)

'For Specific Gravity Option'
If optSpecificGravity.Value = True Then
   For j = 1 To 9999
   Pressure3 = j

   Ppc = 709.604 - (58.718 * SG)
   Tpc = 170.491 + (307.344 * SG)

   Ma = SG * 29

      Ppr = Pressure3 / Ppc
      Tpr = T / Tpc

'Z-Factor (SG)'

   If Ppr > 0.2 And Ppr < 1.2 Then
      If Tpr > 1.05 And Tpr < 1.2 Then
         z = Ppr * ((1.6643 * Tpr) - 2.2114) - (0.3647 * Tpr) + 1.4385
      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
         z = Ppr * ((0.5222 * Tpr) - 0.8511) - (0.0364 * Tpr) + 1.049
      ElseIf Tpr > 1.4 And Tpr < 2# Then
         z = Ppr * ((0.1391 * Tpr) - 0.2988) + (0.0007 * Tpr) + 0.9969
      ElseIf Tpr > 2# And Tpr < 3# Then
         z = Ppr * ((0.0295 * Tpr) - 0.0825) + (0.0009 * Tpr) + 0.9967
      End If
   ElseIf Ppr > 1.2 And Ppr < 2.8 Then
      If Tpr > 1.05 And Tpr < 1.2 Then
         z = Ppr * ((-1.357 * Tpr) + 1.4942) + (4.6315 * Tpr) - 4.7009
      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
         z = Ppr * ((0.1717 * Tpr) - 0.3232) + (0.5869 * Tpr) + 0.1229
      ElseIf Tpr > 1.4 And Tpr < 2# Then
         z = Ppr * ((0.0984 * Tpr) - 0.2053) + (0.0621 * Tpr) + 0.858
      ElseIf Tpr > 2# And Tpr < 3# Then
         z = Ppr * ((0.0211 * Tpr) - 0.0527) + (0.0127 * Tpr) + 0.9549
      End If
   ElseIf Ppr > 2.8 And Ppr < 5.4 Then
      If Tpr > 1.05 And Tpr < 1.2 Then
         z = Ppr * ((-0.3278 * Tpr) + 0.4752) + (1.8223 * Tpr) - 1.9036
      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
         z = Ppr * ((-0.2521 * Tpr) + 0.3871) + (1.6087 * Tpr) - 1.6635
      ElseIf Tpr > 1.4 And Tpr < 2# Then
         z = Ppr * ((-0.0284 * Tpr) + 0.0625) + (0.4714 * Tpr) - 0.0011
      ElseIf Tpr > 2# And Tpr < 3# Then
         z = Ppr * ((0.0041 * Tpr) + 0.0039) + (0.0607 * Tpr) + 0.7927
      End If
   ElseIf Ppr > 5.4 And Ppr < 15# Then
      If Tpr > 1.05 And Tpr < 3# Then
         z = Ppr * ((0.711 + (3.66 * Tpr)) ^ (-1.4667)) - (1.637 / ((0.319 * Tpr) + 0.522)) + 2.071
      End If
   ElseIf Ppr > 15# Then
         z = 1
   ElseIf Ppr < 0.2 Then
         z = 1
   End If
```

```
'Compressibility (SG)'
A1 = 0.31506237
A2 = -1.0467099
A3 = -0.57832729
A4 = 0.53530771
A5 = -0.61232032
A6 = -0.10488813
A7 = 0.68157001
A8 = 0.68446549

   rho = 0.27 * (Ppr / (z * Tpr))

   D = (A1 + (A2 / Tpr) + (A3 / (Tpr ^ 3)))
   E = (2 * (A4 + (A5 / Tpr)) * rho)
   F = (5 * (A5) * (A6) * (rho ^ 4) / Tpr)
   G = (2 * (A7) * (rho) / (Tpr ^ 3))
   H = (1 + (A8 * (rho ^ 2)) - ((A8 ^ 2) * (rho ^ 4)))
   I2 = (Exp(-A8 * (rho ^ 2)))

   dzdrho = D + E + F + (G * H * I2)

   Cpr = (1 / Ppr) - ((0.27 / (z ^ 2 * Tpr)) * ((dzdrho) / (1 + (rho / z) * dzdrho)))

   Cg = Cpr / Ppc

'Viscosity (SG)'

   rhog = ((2.7 * (SG) * (Pressure3)) / ((z) * (T))) * 0.016018
   K = ((9.4 + (0.02 * Ma)) * (T ^ 1.5)) / (209 + (19 * Ma) + T)
   X = 3.5 + (986 / T) + (0.01 * Ma)
   Y = 2.4 - (0.2 * X)
   Viscosity = (K * 10 ^ (-4)) * Exp(X * (rhog ^ Y))

If Pressure3 = PAverage Then
   ViscosityAvg = Viscosity
   CgAvg = Cg
   zAvg = z
   Bgavg = (0.00504 * ((z * T) / Pressure3)) * 1000
End If

If Pressure3 = Pinitial Then
   PseudoPT = PT / (Viscosity * Cg)
   txtPseudoPT.Text = Format(Val(PseudoPT), "#")
   PTAdj = PseudoPT * (ViscosityAvg * CgAvg)
   txtPTAdj.Text = Format(Val(PTAdj), "#")
End If

Next j

ElseIf optGasComposition = True Then
For j = 1 To 9999
Pressure3 = j

C1 = (Val(txtC1.Text)) / 100
M1 = C1 * (16.043)
Tc1 = C1 * (-116.67 + 460#)
Pc1 = C1 * (666.4)
```

```
C2 = (Val(txtC2.Text)) / 100
M2 = C2 * (30.07)
Tc2 = C2 * (89.92 + 460#)
Pc2 = C2 * (706.5)

C3 = (Val(txtC3.Text)) / 100
M3 = C3 * (44.097)
Tc3 = C3 * (206.06 + 460#)
Pc3 = C3 * (616#)

C6 = (Val(txtC6.Text)) / 100
M6 = C6 * (86.177)
Tc6 = C6 * (453.6 + 460#)
Pc6 = C6 * (436.9)

C7 = (Val(txtC7.Text)) / 100
M7 = C7 * (100.204)
Tc7 = C7 * (512.7 + 460#)
Pc7 = C7 * (396.8)

If Val(txtC7plus.Text) > 0 Then
C7plus = (Val(txtC7plus.Text)) / 100
M7p = Val(txtM7p.Text)
M7plus = C7plus * M7p
SG7plus = Val(txtSG7plus.Text)

Tb = (4.5579 * (M7p ^ 0.15178) * (SG7plus ^ 0.15427)) ^ 3

Tc7p = 341.7 + (811 * SG7plus) + ((0.4244 + (0.1174 * SG7plus)) * Tb) + ((0.4669 - (3.2623 * SG7plus))
* (10 ^ 5) / Tb)

A = (0.24244 + (2.2898 / SG7plus) + (0.11857 / (SG7plus ^ 2))) * (0.001) * Tb
B = (1.4685 + (3.648 / SG7plus) + (0.47227 / (SG7plus ^ 2))) * (0.0000001) * (Tb ^ 2)
C = (0.42019 + (1.6977 / (SG7plus ^ 2)) * (0.0000000001) * (Tb ^ 3))
Pc7p = Exp(8.3634 - (0.0566 / SG7plus) - A + B - C)

Tc7plus = C7plus * Tc7p
Pc7plus = C7plus * Pc7p
Else

Tc7plus = 0#
Pc7plus = 0#
M7plus = 0#
End If

CO2 = (Val(txtCO2.Text)) / 100
MCO2 = CO2 * (44.01)
TcCO2 = CO2 * (87.91 + 460#)
PcCO2 = CO2 * (1071#)

H2S = (Val(txtH2S.Text)) / 100
MH2S = H2S * (34.08)
TcH2S = H2S * (212.45 + 460#)
PcH2S = H2S * (1300#)

IC4 = (Val(txtIC4.Text)) / 100
MI4 = IC4 * (58.123)
TcI4 = IC4 * (274.46 + 460#)
```

```
PcI4 = IC4 * (527.9)

IC5 = (Val(txtIC5.Text)) / 100
MI5 = IC5 * (72.15)
TcI5 = IC5 * (369.1 + 460#)
PcI5 = IC5 * (490.4)

N2 = (Val(txtN2.Text)) / 100
MN2 = N2 * (28.0134)
TcN2 = N2 * (-232.51 + 460#)
PcN2 = N2 * (493.1)

NC4 = (Val(txtNC4.Text)) / 100
MN4 = NC4 * (58.123)
TcN4 = NC4 * (305.62 + 460#)
PcN4 = NC4 * (550.6)

NC5 = (Val(txtNC5.Text)) / 100
MN5 = NC5 * (72.15)
TcN5 = NC5 * (385.8 + 460#)
PcN5 = NC5 * (488.6)

C8 = (Val(txtC8.Text)) / 100
M8 = C8 * (114.231)
Tc8 = C8 * (564.22 + 460#)
Pc8 = C8 * (360.7)

C9 = (Val(txtC9.Text)) / 100
M9 = C9 * (128.258)
Tc9 = C9 * (610.68 + 460#)
Pc9 = C9 * (331.8)

C10 = (Val(txtC10.Text)) / 100
M10 = C10 * (142.285)
Tc10 = C10 * (652# + 460#)
Pc10 = C10 * (305.2)

'Critical Properties'
Tpc = Tc1 + Tc2 + Tc3 + TcI4 + TcN4 + TcI5 + TcN5 + Tc6 + Tc7 + Tc7plus + Tc8 + Tc9 + Tc10 +
TcN2 + TcH2S + TcCO2
Ppc = Pc1 + Pc2 + Pc3 + PcI4 + PcN4 + PcI5 + PcN5 + Pc6 + Pc7 + Pc7plus + Pc8 + Pc9 + Pc10 + PcN2
+ PcH2S + PcCO2
Ma = M1 + M2 + M3 + MI4 + MN4 + MI5 + MN5 + M6 + M7 + M7plus + M8 + M9 + M10 + MN2 +
MH2S + MCO2

SG = Ma / 29

'Correction for Critical Properties'
corr = (120 * (((H2S + CO2) ^ 0.9) - ((CO2 + H2S) ^ 1.6))) + (15 * ((H2S ^ 0.5) - (H2S ^ 4#)))
Tpccorr = Tpc - corr
Ppccorr = (Ppc * Tpccorr) / (Tpc + ((H2S / 100) * (1 - (H2S / 100)) * corr))

'Reduced Properties'
   Ppr = Pressure3 / Ppccorr
   Tpr = T / Tpccorr

'Z-Factor Calculation'
If Ppr > 0.2 And Ppr < 1.2 Then
```

```
        If Tpr > 1.05 And Tpr < 1.2 Then
            z = Ppr * ((1.6643 * Tpr) - 2.2114) - (0.3647 * Tpr) + 1.4385
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((0.5222 * Tpr) - 0.8511) - (0.0364 * Tpr) + 1.049
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((0.1391 * Tpr) - 0.2988) + (0.0007 * Tpr) + 0.9969
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0295 * Tpr) - 0.0825) + (0.0009 * Tpr) + 0.9967
        End If
    ElseIf Ppr > 1.2 And Ppr < 2.8 Then
        If Tpr > 1.05 And Tpr < 1.2 Then
            z = Ppr * ((-1.357 * Tpr) + 1.4942) + (4.6315 * Tpr) - 4.7009
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((0.1717 * Tpr) - 0.3232) + (0.5869 * Tpr) + 0.1229
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((0.0984 * Tpr) - 0.2053) + (0.0621 * Tpr) + 0.858
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0211 * Tpr) - 0.0527) + (0.0127 * Tpr) + 0.9549
        End If
    ElseIf Ppr > 2.8 And Ppr < 5.4 Then
        If Tpr > 1.05 And Tpr < 1.2 Then
            z = Ppr * ((-0.3278 * Tpr) + 0.4752) + (1.8223 * Tpr) - 1.9036
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((-0.2521 * Tpr) + 0.3871) + (1.6087 * Tpr) - 1.6635
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((-0.0284 * Tpr) + 0.0625) + (0.4714 * Tpr) - 0.0011
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0041 * Tpr) + 0.0039) + (0.0607 * Tpr) + 0.7927
        End If
    ElseIf Ppr > 5.4 And Ppr < 15# Then
        If Tpr > 1.05 And Tpr < 3# Then
            z = Ppr * ((0.711 + (3.66 * Tpr)) ^ (-1.4667)) - (1.637 / ((0.319 * Tpr) + 0.522)) + 2.071
        End If
    ElseIf Ppr > 15# Then
            z = 1
    ElseIf Ppr < 0.2 Then
            z = 1
    End If

'Compressibility'
A1 = 0.31506237
A2 = -1.0467099
A3 = -0.57832729
A4 = 0.53530771
A5 = -0.61232032
A6 = -0.10488813
A7 = 0.68157001
A8 = 0.68446549

    rho = (0.27 * Ppr) / (z * Tpr)

    D = (A1 + (A2 / Tpr) + (A3 / (Tpr ^ 3)))
    E = (2 * (A4 + (A5 / Tpr)) * rho)
    F = (5 * (A5) * (A6) * (rho ^ 4) / (Tpr))
    G = ((2 * (A7) * (rho)) / (Tpr ^ 3))
    H = (1 + (A8 * (rho ^ 2)) - ((A8 ^ 2) * (rho ^ 4)))
    I2 = (Exp((-A8) * (rho ^ 2)))
```

```
   dzdrho = D + E + F + (G * H * I2)

   Cpr = (1 / Ppr) - ((0.27 / (z ^ 2 * Tpr)) * ((dzdrho) / (1 + (rho / z) * dzdrho)))

   Cg = Cpr / Ppc

'Viscosity'

   rhog = ((2.7 * (SG) * (Pressure3)) / ((z) * (T))) * 0.016018
   K = ((9.4 + (0.02 * Ma)) * (T ^ 1.5)) / (209# + (19# * Ma) + T)
   X = 3.5 + (986# / T) + (0.01 * Ma)
   Y = 2.4 - (0.2 * X)
   Viscosity = (K * 10 ^ (-4)) * Exp(X * (rhog ^ Y))

If Pressure3 = PAverage Then
   ViscosityAvg = Viscosity
   CgAvg = Cg
   zAvg = z
   Bgavg = (0.00504 * ((z * T) / Pressure3)) * 1000
End If

If Pressure3 = Pinitial Then
   PseudoPT = PT / (Viscosity * Cg)
   txtPseudoPT.Text = Format(Val(PseudoPT), "#")
   PTAdj = PseudoPT * (ViscosityAvg * CgAvg)
   txtPTAdj.Text = Format(Val(PTAdj), "#")
End If
Next j


End If
End Sub

Public Sub txtTemp_Change()
degrees_F = Val(txtTemp.Text)
T = degrees_F + 460#
End Sub

Public Sub Form_Load()
ReDim Preserve Pressure(10000) As Double
ReDim Preserve dPmainavg(10000) As Double
ReDim Preserve mainavg(10000) As Double
ReDim Preserve PseudoPress(10000) As Double
ReDim Preserve P(100) As Double
ReDim Preserve SITime(100) As Double
ReDim Preserve main(10000) As Double
ReDim Preserve Ai(100) As Double
ReDim Preserve Ci(100) As Double
ReDim Preserve Tan(100) As Double
ReDim Preserve Viscosity2(10000) As Double
ReDim Preserve Cg2(10000) As Double
'ReDim Preserve Ppr(100) As Double
'ReDim Preserve rho(100) As Double
'ReDim Preserve E(100) As Double
'ReDim Preserve F(100) As Double
'ReDim Preserve G(100) As Double
'ReDim Preserve H(100) As Double
'ReDim Preserve I2(100) As Double
```

```
'ReDim Preserve dzdrho(100) As Double
'ReDim Preserve Cpr(100) As Double
'ReDim Preserve Cg(100) As Double
'ReDim Preserve rhog(100) As Double
'ReDim Preserve z(100) As Double
'ReDim Preserve Viscosity(100) As Double

fraGasComposition.Visible = False
End Sub

Public Sub optGasComposition_Click()
fraGasComposition.Visible = True
Image1.Visible = False
End Sub

Public Sub optSpecificGravity_Click()
fraGasComposition.Visible = False
Image1.Visible = True
SG = Val(InputBox("Enter the Gas Gravity:", "Gas Gravity"))
End Sub



Public Sub cmdCalculate_Click()

For j = 1 To counter
   If P(j) > P(j - 1) Then
   maxP = P(j)
   End If
Next j
maxP = maxP

For j = 1 To maxP
   Pressure(j) = j
Next j

frmResults2.grdResults.Rows = counter

Open "a:results.txt" For Output As #2
Print #2, "   P", "  Tpr", "  Ppr", " Z-Factor", "    Cg", " Viscosity", "(2P)/((mhu)*z)", "(2P)/((mhu)*z)avg",
"Pseudo-Pressure", "       Adjusted Pressure", "    SI time", "        Horner Time", "                 Ci", "
Pseudo-Time", "          Pseudo Horner Time", "        Adjusted Time", "  Adj. Horner Time",

degrees_F = Val(txtTemp.Text)
T = degrees_F + 460#

frmResults2.Show

'For Specific Gravity Option'
If optSpecificGravity.Value = True Then

  For j = 1 To maxP
   Pressure2 = Pressure(j)

   Ppc = 709.604 - (58.718 * SG)
   Tpc = 170.491 + (307.344 * SG)

   Ma = SG * 29
```

```
      Ppr = Pressure(j) / Ppc
      Tpr = T / Tpc

'Z-Factor (SG)'

   If Ppr > 0.2 And Ppr < 1.2 Then
      If Tpr > 1.05 And Tpr < 1.2 Then
         z = Ppr * ((1.6643 * Tpr) - 2.2114) - (0.3647 * Tpr) + 1.4385
      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
         z = Ppr * ((0.5222 * Tpr) - 0.8511) - (0.0364 * Tpr) + 1.049
      ElseIf Tpr > 1.4 And Tpr < 2# Then
         z = Ppr * ((0.1391 * Tpr) - 0.2988) + (0.0007 * Tpr) + 0.9969
      ElseIf Tpr > 2# And Tpr < 3# Then
         z = Ppr * ((0.0295 * Tpr) - 0.0825) + (0.0009 * Tpr) + 0.9967
      End If
   ElseIf Ppr > 1.2 And Ppr < 2.8 Then
      If Tpr > 1.05 And Tpr < 1.2 Then
         z = Ppr * ((-1.357 * Tpr) + 1.4942) + (4.6315 * Tpr) - 4.7009
      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
         z = Ppr * ((0.1717 * Tpr) - 0.3232) + (0.5869 * Tpr) + 0.1229
      ElseIf Tpr > 1.4 And Tpr < 2# Then
         z = Ppr * ((0.0984 * Tpr) - 0.2053) + (0.0621 * Tpr) + 0.858
      ElseIf Tpr > 2# And Tpr < 3# Then
         z = Ppr * ((0.0211 * Tpr) - 0.0527) + (0.0127 * Tpr) + 0.9549
      End If
   ElseIf Ppr > 2.8 And Ppr < 5.4 Then
      If Tpr > 1.05 And Tpr < 1.2 Then
         z = Ppr * ((-0.3278 * Tpr) + 0.4752) + (1.8223 * Tpr) - 1.9036
      ElseIf Tpr > 1.2 And Tpr < 1.4 Then
         z = Ppr * ((-0.2521 * Tpr) + 0.3871) + (1.6087 * Tpr) - 1.6635
      ElseIf Tpr > 1.4 And Tpr < 2# Then
         z = Ppr * ((-0.0284 * Tpr) + 0.0625) + (0.4714 * Tpr) - 0.0011
      ElseIf Tpr > 2# And Tpr < 3# Then
         z = Ppr * ((0.0041 * Tpr) + 0.0039) + (0.0607 * Tpr) + 0.7927
      End If
   ElseIf Ppr > 5.4 And Ppr < 15# Then
      If Tpr > 1.05 And Tpr < 3# Then
         z = Ppr * ((0.711 + (3.66 * Tpr)) ^ (-1.4667)) - (1.637 / ((0.319 * Tpr) + 0.522)) + 2.071
      End If
   ElseIf Ppr > 15# Then
         z = 1
   ElseIf Ppr < 0.2 Then
         z = 1
   End If

'Compressibility (SG)'
A1 = 0.31506237
A2 = -1.0467099
A3 = -0.57832729
A4 = 0.53530771
A5 = -0.61232032
A6 = -0.10488813
A7 = 0.68157001
A8 = 0.68446549

   rho = 0.27 * (Ppr / (z * Tpr))
```

```
    D = (A1 + (A2 / Tpr) + (A3 / (Tpr ^ 3)))
    E = (2 * (A4 + (A5 / Tpr)) * rho)
    F = (5 * (A5) * (A6) * (rho ^ 4) / Tpr)
    G = (2 * (A7) * (rho) / (Tpr ^ 3))
    H = (1 + (A8 * (rho ^ 2)) - ((A8 ^ 2) * (rho ^ 4)))
    I2 = (Exp(-A8 * (rho ^ 2)))

    dzdrho = D + E + F + (G * H * I2)

    Cpr = (1 / Ppr) - ((0.27 / (z ^ 2 * Tpr)) * ((dzdrho) / (1 + (rho / z) * dzdrho)))

    Cg = Cpr / Ppc

'Viscosity (SG)'

    rhog = ((2.7 * (SG) * (Pressure(j))) / ((z) * (T))) * 0.016018
    K = ((9.4 + (0.02 * Ma)) * (T ^ 1.5)) / (209 + (19 * Ma) + T)
    X = 3.5 + (986 / T) + (0.01 * Ma)
    Y = 2.4 - (0.2 * X)
    Viscosity = (K * 10 ^ (-4)) * Exp(X * (rhog ^ Y))



dPressure = Pressure2 - Pressure(j - 1)
main(j) = ((2 * Pressure2) / (Viscosity * z))
mainavg(j) = (main(j) + main(j - 1)) / 2
dPmainavg(j) = mainavg(j) * dPressure
PseudoPress(j) = dPmainavg(j) + PseudoPress(j - 1)

PressureAdj = PseudoPress(j) * ((ViscosityAvg * zAvg) / (2 * PAverage))

If Pressure2 = P(1) Then
    Pawf = PressureAdj
End If

For i = 1 To counter

  If Pressure2 = P(i) Then
    frmResults2.grdResults.Col = 1
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Pressure2), "###,###")

    frmResults2.grdResults.Col = 2
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Tpr), "0.00")

    frmResults2.grdResults.Col = 3
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Ppr), "0.00")

    frmResults2.grdResults.Col = 4
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(z), "0.00")
```

```
    frmResults2.grdResults.Col = 5
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Cg), "0.00000000")

    frmResults2.grdResults.Col = 6
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Viscosity), "0.0000000")

    frmResults2.grdResults.Col = 7
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(main(j)), "###,##0")

    frmResults2.grdResults.Col = 8
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(mainavg(j)), "###,##0")

    frmResults2.grdResults.Col = 9
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(PseudoPress(j)), "###,###,##0")

    frmResults2.grdResults.Col = 10
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(PressureAdj), "###,##0")

    Ai(i) = 1 / (Viscosity * Cg)
    Bi = (Ai(i) + Ai(i - 1)) / 2
    Ci(i) = Bi * (SITime(i) - SITime(i - 1))

    Tan(i) = Ci(i) + Tan(i - 1)

    TimeAdj = Tan(i) * (ViscosityAvg * CgAvg)

    If SITime(i) = 0 Then
    HornerTime = 0
    HornerTimeAdj = 0
    GoTo 8
    Else
    HornerTime = (PT + SITime(i)) / (SITime(i))
    HornerTimeAdj = (PTAdj + TimeAdj) / (TimeAdj)
    PseudoHornerTime = (PseudoPT + Tan(i)) / (Tan(i))
8:
    End If

    frmResults2.grdResults.Col = 11
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(SITime(i)), "###,##0.00")

    frmResults2.grdResults.Col = 12
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(HornerTime), "###,##0.00")
```

```
    frmResults2.grdResults.Col = 13
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Ci(i)), "###,###,##0")

    frmResults2.grdResults.Col = 14
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Tan(i)), "###,###,##0")

    frmResults2.grdResults.Col = 15
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(PseudoHornerTime), "###,##0.0")

    frmResults2.grdResults.Col = 16
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(TimeAdj), "0.000")

    frmResults2.grdResults.Col = 17
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(HornerTimeAdj), "###,##0.0")

    Print #2, Format(Val(Pressure2), "0"), Format(Val(Tpr), "0.00"), Format(Val(Ppr), "0.00"),
Format(Val(z), "0.00"), Format(Val(Cg), "0.000000"), Format(Val(Viscosity), "0.00000"),
Format(Val(main(j)), "          0"), Format(Val(mainavg(j)), "          0"), Format(Val(PseudoPress(j)), "
0.00"), Format(Val(PressureAdj), "                0"), Format(Val(SITime(i)), "        0.00"),
Format(Val(HornerTime), "                0.00"), Format(Val(Ci(i)), "                0"), Format(Val(Tan(i)), "
0"), Format(Val(PseudoHornerTime), "0.0"), Format(Val(TimeAdj), "                0.000"),
Format(Val(HornerTimeAdj), "                0.0")
  End If
Next i
frmzfactor2.Hide
Next j
Close #2

ElseIf optGasComposition = True Then
C1 = (Val(txtC1.Text)) / 100
M1 = C1 * (16.043)
Tc1 = C1 * (-116.67 + 460#)
Pc1 = C1 * (666.4)

C2 = (Val(txtC2.Text)) / 100
M2 = C2 * (30.07)
Tc2 = C2 * (89.92 + 460#)
Pc2 = C2 * (706.5)

C3 = (Val(txtC3.Text)) / 100
M3 = C3 * (44.097)
Tc3 = C3 * (206.06 + 460#)
Pc3 = C3 * (616#)

C6 = (Val(txtC6.Text)) / 100
M6 = C6 * (86.177)
Tc6 = C6 * (453.6 + 460#)
```

```
Pc6 = C6 * (436.9)

C7 = (Val(txtC7.Text)) / 100
M7 = C7 * (100.204)
Tc7 = C7 * (512.7 + 460#)
Pc7 = C7 * (396.8)

If Val(txtC7plus.Text) > 0 Then
C7plus = (Val(txtC7plus.Text)) / 100
M7p = Val(txtM7p.Text)
M7plus = C7plus * M7p
SG7plus = Val(txtSG7plus.Text)

Tb = (4.5579 * (M7p ^ 0.15178) * (SG7plus ^ 0.15427)) ^ 3

Tc7p = 341.7 + (811 * SG7plus) + ((0.4244 + (0.1174 * SG7plus)) * Tb) + ((0.4669 - (3.2623 * SG7plus))
* (10 ^ 5) / Tb)

A = (0.24244 + (2.2898 / SG7plus) + (0.11857 / (SG7plus ^ 2))) * (0.001) * Tb
B = (1.4685 + (3.648 / SG7plus) + (0.47227 / (SG7plus ^ 2))) * (0.0000001) * (Tb ^ 2)
C = (0.42019 + (1.6977 / (SG7plus ^ 2)) * (0.0000000001) * (Tb ^ 3))
Pc7p = Exp(8.3634 - (0.0566 / SG7plus) - A + B - C)

Tc7plus = C7plus * Tc7p
Pc7plus = C7plus * Pc7p
Else

Tc7plus = 0#
Pc7plus = 0#
M7plus = 0#
End If

CO2 = (Val(txtCO2.Text)) / 100
MCO2 = CO2 * (44.01)
TcCO2 = CO2 * (87.91 + 460#)
PcCO2 = CO2 * (1071#)

H2S = (Val(txtH2S.Text)) / 100
MH2S = H2S * (34.08)
TcH2S = H2S * (212.45 + 460#)
PcH2S = H2S * (1300#)

IC4 = (Val(txtIC4.Text)) / 100
MI4 = IC4 * (58.123)
TcI4 = IC4 * (274.46 + 460#)
PcI4 = IC4 * (527.9)

IC5 = (Val(txtIC5.Text)) / 100
MI5 = IC5 * (72.15)
TcI5 = IC5 * (369.1 + 460#)
PcI5 = IC5 * (490.4)

N2 = (Val(txtN2.Text)) / 100
MN2 = N2 * (28.0134)
TcN2 = N2 * (-232.51 + 460#)
PcN2 = N2 * (493.1)

NC4 = (Val(txtNC4.Text)) / 100
```

```
MN4 = NC4 * (58.123)
TcN4 = NC4 * (305.62 + 460#)
PcN4 = NC4 * (550.6)

NC5 = (Val(txtNC5.Text)) / 100
MN5 = NC5 * (72.15)
TcN5 = NC5 * (385.8 + 460#)
PcN5 = NC5 * (488.6)

C8 = (Val(txtC8.Text)) / 100
M8 = C8 * (114.231)
Tc8 = C8 * (564.22 + 460#)
Pc8 = C8 * (360.7)

C9 = (Val(txtC9.Text)) / 100
M9 = C9 * (128.258)
Tc9 = C9 * (610.68 + 460#)
Pc9 = C9 * (331.8)

C10 = (Val(txtC10.Text)) / 100
M10 = C10 * (142.285)
Tc10 = C10 * (652# + 460#)
Pc10 = C10 * (305.2)

'Critical Properties'
Tpc = Tc1 + Tc2 + Tc3 + TcI4 + TcN4 + TcI5 + TcN5 + Tc6 + Tc7 + Tc7plus + Tc8 + Tc9 + Tc10 +
TcN2 + TcH2S + TcCO2
Ppc = Pc1 + Pc2 + Pc3 + PcI4 + PcN4 + PcI5 + PcN5 + Pc6 + Pc7 + Pc7plus + Pc8 + Pc9 + Pc10 + PcN2
+ PcH2S + PcCO2
Ma = M1 + M2 + M3 + MI4 + MN4 + MI5 + MN5 + M6 + M7 + M7plus + M8 + M9 + M10 + MN2 +
MH2S + MCO2

SG = Ma / 29

'Correction for Critical Properties'
corr = (120 * (((H2S + CO2) ^ 0.9) - ((CO2 + H2S) ^ 1.6))) + (15 * ((H2S ^ 0.5) - (H2S ^ 4#)))
Tpccorr = Tpc - corr
Ppccorr = (Ppc * Tpccorr) / (Tpc + ((H2S / 100) * (1 - (H2S / 100)) * corr))

'Reduced Properties'
For j = 1 To maxP
  Pressure2 = Pressure(j)
  Ppr = Pressure(j) / Ppccorr
  Tpr = T / Tpccorr

'Z-Factor Calculation'
If Ppr > 0.2 And Ppr < 1.2 Then
    If Tpr > 1.05 And Tpr < 1.2 Then
      z = Ppr * ((1.6643 * Tpr) - 2.2114) - (0.3647 * Tpr) + 1.4385
    ElseIf Tpr > 1.2 And Tpr < 1.4 Then
      z = Ppr * ((0.5222 * Tpr) - 0.8511) - (0.0364 * Tpr) + 1.049
    ElseIf Tpr > 1.4 And Tpr < 2# Then
      z = Ppr * ((0.1391 * Tpr) - 0.2988) + (0.0007 * Tpr) + 0.9969
    ElseIf Tpr > 2# And Tpr < 3# Then
      z = Ppr * ((0.0295 * Tpr) - 0.0825) + (0.0009 * Tpr) + 0.9967
    End If
  ElseIf Ppr > 1.2 And Ppr < 2.8 Then
    If Tpr > 1.05 And Tpr < 1.2 Then
```

```
            z = Ppr * ((-1.357 * Tpr) + 1.4942) + (4.6315 * Tpr) - 4.7009
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((0.1717 * Tpr) - 0.3232) + (0.5869 * Tpr) + 0.1229
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((0.0984 * Tpr) - 0.2053) + (0.0621 * Tpr) + 0.858
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0211 * Tpr) - 0.0527) + (0.0127 * Tpr) + 0.9549
        End If
    ElseIf Ppr > 2.8 And Ppr < 5.4 Then
        If Tpr > 1.05 And Tpr < 1.2 Then
            z = Ppr * ((-0.3278 * Tpr) + 0.4752) + (1.8223 * Tpr) - 1.9036
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((-0.2521 * Tpr) + 0.3871) + (1.6087 * Tpr) - 1.6635
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((-0.0284 * Tpr) + 0.0625) + (0.4714 * Tpr) - 0.0011
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0041 * Tpr) + 0.0039) + (0.0607 * Tpr) + 0.7927
        End If
    ElseIf Ppr > 5.4 And Ppr < 15# Then
        If Tpr > 1.05 And Tpr < 3# Then
            z = Ppr * ((0.711 + (3.66 * Tpr)) ^ (-1.4667)) - (1.637 / ((0.319 * Tpr) + 0.522)) + 2.071
        End If
    ElseIf Ppr > 15# Then
            z = 1
    ElseIf Ppr < 0.2 Then
            z = 1
    End If

'Compressibility'
A1 = 0.31506237
A2 = -1.0467099
A3 = -0.57832729
A4 = 0.53530771
A5 = -0.61232032
A6 = -0.10488813
A7 = 0.68157001
A8 = 0.68446549

    rho = (0.27 * Ppr) / (z * Tpr)

    D = (A1 + (A2 / Tpr) + (A3 / (Tpr ^ 3)))
    E = (2 * (A4 + (A5 / Tpr)) * rho)
    F = (5 * (A5) * (A6) * (rho ^ 4) / (Tpr))
    G = ((2 * (A7) * (rho)) / (Tpr ^ 3))
    H = (1 + (A8 * (rho ^ 2)) - ((A8 ^ 2) * (rho ^ 4)))
    I2 = (Exp((-A8) * (rho ^ 2)))

    dzdrho = D + E + F + (G * H * I2)

    Cpr = (1 / Ppr) - ((0.27 / (z ^ 2 * Tpr)) * ((dzdrho) / (1 + (rho / z) * dzdrho)))

    Cg = Cpr / Ppc

'Viscosity'

    rhog = ((2.7 * (SG) * (Pressure(j))) / ((z) * (T))) * 0.016018
    K = ((9.4 + (0.02 * Ma)) * (T ^ 1.5)) / (209# + (19# * Ma) + T)
    X = 3.5 + (986# / T) + (0.01 * Ma)
```

```
    Y = 2.4 - (0.2 * X)
    Viscosity = (K * 10 ^ (-4)) * Exp(X * (rhog ^ Y))

dPressure = Pressure2 - Pressure(j - 1)
main(j) = ((2 * Pressure2) / (Viscosity * z))
mainavg(j) = (main(j) + main(j - 1)) / 2
dPmainavg(j) = mainavg(j) * dPressure
PseudoPress(j) = dPmainavg(j) + PseudoPress(j - 1)

PressureAdj = PseudoPress(j) * ((ViscosityAvg * zAvg) / (2 * PAverage))

frmResults2.Show

For i = 1 To counter

  If Pressure2 = P(i) Then
    frmResults2.grdResults.Col = 1
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Pressure2), "###,##0")

    frmResults2.grdResults.Col = 2
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Tpr), "0.00")

    frmResults2.grdResults.Col = 3
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Ppr), "0.00")

    frmResults2.grdResults.Col = 4
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(z), "0.00")

    frmResults2.grdResults.Col = 5
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Cg), "0.00000000")

    frmResults2.grdResults.Col = 6
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Viscosity), "0.0000000")

    frmResults2.grdResults.Col = 7
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(main(j)), "###,##0")

    frmResults2.grdResults.Col = 8
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(mainavg(j)), "###,##0")

    frmResults2.grdResults.Col = 9
    frmResults2.grdResults.Row = i
```

```
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(PseudoPress(j)), "###,###,##0")

    frmResults2.grdResults.Col = 10
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(PressureAdj), "###,##0")

    Ai(i) = 1 / (Viscosity * Cg)
    Bi = (Ai(i) + Ai(i - 1)) / 2
    Ci(i) = Bi * (SITime(i) - SITime(i - 1))

    Tan(i) = Ci(i) + Tan(i - 1)

    TimeAdj = Tan(i) * (ViscosityAvg * CgAvg)

    If SITime(i) = 0 Then
    HornerTime = 0
    HornerTimeAdj = 0
    GoTo 9
    Else
    HornerTime = (PT + SITime(i)) / (SITime(i))
    HornerTimeAdj = (PTAdj + TimeAdj) / (TimeAdj)
    PseudoHornerTime = (PseudoPT + Tan(i)) / (Tan(i))
9:
    End If

    frmResults2.grdResults.Col = 11
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(SITime(i)), "0.00")

    frmResults2.grdResults.Col = 12
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(HornerTime), "###,##0.00")

    frmResults2.grdResults.Col = 13
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Ci(i)), "###,###")

    frmResults2.grdResults.Col = 14
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(Tan(i)), "###,###")

    frmResults2.grdResults.Col = 15
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(PseudoHornerTime), "###,##0.0")

    frmResults2.grdResults.Col = 16
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(TimeAdj), "0.000")

    frmResults2.grdResults.Col = 17
```

```
    frmResults2.grdResults.Row = i
    frmResults2.grdResults.CellAlignment = 3
    frmResults2.grdResults.Text = Format(Val(HornerTimeAdj), "###,##0.0")

    Print #2, Format(Val(Pressure2), "0"), Format(Val(Tpr), "0.00"), Format(Val(Ppr), "0.00"),
Format(Val(z), "0.00"), Format(Val(Cg), "0.000000"), Format(Val(Viscosity), "0.00000"),
Format(Val(main(j)), "          0"), Format(Val(mainavg(j)), "          0"), Format(Val(PseudoPress(j)), "
0.00"), Format(Val(PressureAdj), "                0"), Format(Val(SITime(i)), "        0.00"),
Format(Val(HornerTime), "                0.00"), Format(Val(Ci(i)), "                0"), Format(Val(Tan(i)), "
0"), Format(Val(PseudoHornerTime), "0.0"), Format(Val(TimeAdj), "                0.000"),
Format(Val(HornerTimeAdj), "                0.0")

  End If
Next i
frmzfactor2.Hide
Next j
Close #2
Else
MsgBox ("Please choose to input either Specific Gravity or the Composition")

End If

End Sub

Public Sub Command1_Click()
End
End Sub
```

Calculations:

```
Private Sub cmdBack_Click()
frmCalculations.Hide
frmResults2.Show
End Sub

Private Sub cmdBg_Click()

End Sub

Private Sub cmdCalculate_Click()
flow = Val(txtflow.Text)
height2 = Val(txtheight2.Text)
Porosity = Val(txtPorosity.Text)
radius = Val(txtradius.Text)
PseudoPressure = Val(txtPseudoPressure.Text)
pseudoslope = Val(txtpseudoslope.Text)
PseudoPressure2 = Val(txtPseudoPressure2.Text)
pseudoslope2 = Val(txtpseudoslope2.Text)
PseudoPress2 = Val(txtPseudoPress2.Text)
PseudoTime2 = Val(txtPseudoTime2.Text)
Pawf = Val(txtPawf.Text)
AdjPws = Val(txtAdjPws.Text)
adjslope = Val(txtadjslope.Text)
Pastar = Val(txtPastar.Text)
PseudoPress1 = Val(txtPseudoPress1.Text)
deltatime = Val(txtdeltatime.Text)
Visc = Val(txtVisc.Text)
Compress = Val(txtCompress.Text)
dta1 = Val(txtdta1.Text)
```

```
dmP1 = Val(txtdmP1.Text)
dtapPaws = Val(txtdtapPaws.Text)
'tstarpseudo = Val(txttstarpseudo.Text)
'Ptstarpseudo = Val(txtPtstarpseudo.Text)

Pstcond = 14.7
Tstcond = 520#

'Pseudo-Pressure, Time'
PermPseudo = -(1637 * flow * T) / (pseudoslope * height2)
txtPermPseudo.Text = Format(PermPseudo, "0.00")

SkinPseudo = 1.151 * (((PseudoPressure - PseudoPress1) / pseudoslope) - ((Log((PermPseudo * deltatime)
/ (Porosity * Visc * Compress * (radius ^ 2)))) / (Log(10#))) + 3.23)
txtSkinPseudo.Text = Format(SkinPseudo, "0.00")

'WBSPseudo = ((flow) * (Bg) * (Pseudotimestar)) / (24 * DPPseudo)
'txtWBSPseudo.Text = Format(Val(WBSPseudo), "0.00")

'Pseudo-Pressure, Psuedo-Time'
PermPseudo2 = -(1637 * flow * T) / (pseudoslope2 * height2)
txtPermPseudo2.Text = Format(PermPseudo2, "0.00")

SkinPseudo2 = 1.151 * (((PseudoPress2 - PseudoPressure2) / pseudoslope2) - ((Log((PermPseudo2 *
PseudoTime2) / (Porosity * (radius ^ 2)))) / (Log(10#))) + 3.23 + ((Log((PseudoPT + PseudoTime2) /
(PseudoPT))) / (Log(10#))))
txtSkinPseudo2.Text = Format(SkinPseudo2, "0.00")

delta = (dta1 / dmP1)
WBSPseudo = ((13.26 * (flow) * (Pstcond) * (T)) / ((Porosity) * (height2) * (radius ^ 2) * Tstcond)) * delta
txtWBSPseudo.Text = Format(Val(WBSPseudo), "0.00")

'WBStrial = (flow*Bg

'Adjusted Pressure, Adjusted Time'
PermAdj = -(162.6 * (flow) * (Bgavg) * (ViscosityAvg)) / (adjslope * height2)
txtPermAdj.Text = Format(PermAdj, "0.00")

SkinAdj = 1.151 * (((Pawf - AdjPws) / adjslope) - ((Log((PermAdj * dtapPaws) / (Porosity * ViscosityAvg
* CgAvg * (radius ^ 2)))) / (Log(10#))) + 3.23 + ((Log((PTAdj + dtapPaws) / (PTAdj))) / (Log(10#))))
txtSkinAdj.Text = Format(SkinAdj, "0.00")

'WBSAdj = ((flow) * (Bgavg) * (Pseudotimestar)) / (24 * DPPseudo)
'txtWBSAdj.Text = Format(Val(WBSAdj), "0.00")

End Sub

Private Sub cmdCmhu_Click()
'For Specific Gravity Option'
If frmzfactor2.optSpecificGravity.Value = True Then

    Pseudostar = Val(txtPseudostar.Text)

    Ppc = 709.604 - (58.718 * SG)
    Tpc = 170.491 + (307.344 * SG)

    Ma = SG * 29
```

```
        Ppr = Pseudostar / Ppc
        Tpr = T / Tpc

'Z-Factor (SG)'

    If Ppr > 0.2 And Ppr < 1.2 Then
        If Tpr > 1.05 And Tpr < 1.2 Then
            z = Ppr * ((1.6643 * Tpr) - 2.2114) - (0.3647 * Tpr) + 1.4385
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((0.5222 * Tpr) - 0.8511) - (0.0364 * Tpr) + 1.049
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((0.1391 * Tpr) - 0.2988) + (0.0007 * Tpr) + 0.9969
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0295 * Tpr) - 0.0825) + (0.0009 * Tpr) + 0.9967
        End If
    ElseIf Ppr > 1.2 And Ppr < 2.8 Then
        If Tpr > 1.05 And Tpr < 1.2 Then
            z = Ppr * ((-1.357 * Tpr) + 1.4942) + (4.6315 * Tpr) - 4.7009
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((0.1717 * Tpr) - 0.3232) + (0.5869 * Tpr) + 0.1229
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((0.0984 * Tpr) - 0.2053) + (0.0621 * Tpr) + 0.858
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0211 * Tpr) - 0.0527) + (0.0127 * Tpr) + 0.9549
        End If
    ElseIf Ppr > 2.8 And Ppr < 5.4 Then
        If Tpr > 1.05 And Tpr < 1.2 Then
            z = Ppr * ((-0.3278 * Tpr) + 0.4752) + (1.8223 * Tpr) - 1.9036
        ElseIf Tpr > 1.2 And Tpr < 1.4 Then
            z = Ppr * ((-0.2521 * Tpr) + 0.3871) + (1.6087 * Tpr) - 1.6635
        ElseIf Tpr > 1.4 And Tpr < 2# Then
            z = Ppr * ((-0.0284 * Tpr) + 0.0625) + (0.4714 * Tpr) - 0.0011
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0041 * Tpr) + 0.0039) + (0.0607 * Tpr) + 0.7927
        End If
    ElseIf Ppr > 5.4 And Ppr < 15# Then
        If Tpr > 1.05 And Tpr < 3# Then
            z = Ppr * ((0.711 + (3.66 * Tpr)) ^ (-1.4667)) - (1.637 / ((0.319 * Tpr) + 0.522)) + 2.071
        End If
    ElseIf Ppr > 15# Then
            z = 1
    ElseIf Ppr < 0.2 Then
            z = 1
    End If

'Compressibility (SG)'
A1 = 0.31506237
A2 = -1.0467099
A3 = -0.57832729
A4 = 0.53530771
A5 = -0.61232032
A6 = -0.10488813
A7 = 0.68157001
A8 = 0.68446549

    rho = 0.27 * (Ppr / (z * Tpr))

    D = (A1 + (A2 / Tpr) + (A3 / (Tpr ^ 3)))
```

```
   E = (2 * (A4 + (A5 / Tpr)) * rho)
   F = (5 * (A5) * (A6) * (rho ^ 4) / Tpr)
   G = (2 * (A7) * (rho) / (Tpr ^ 3))
   H = (1 + (A8 * (rho ^ 2)) - ((A8 ^ 2) * (rho ^ 4)))
   I2 = (Exp(-A8 * (rho ^ 2)))

   dzdrho = D + E + F + (G * H * I2)

   Cpr = (1 / Ppr) - ((0.27 / (z ^ 2 * Tpr)) * ((dzdrho) / (1 + (rho / z) * dzdrho)))

   Cg = Cpr / Ppc

'Viscosity (SG)'

   rhog = ((2.7 * (SG) * (Pseudostar)) / ((z) * (T))) * 0.016018
   K = ((9.4 + (0.02 * Ma)) * (T ^ 1.5)) / (209 + (19 * Ma) + T)
   X = 3.5 + (986 / T) + (0.01 * Ma)
   Y = 2.4 - (0.2 * X)
   Viscosity = (K * 10 ^ (-4)) * Exp(X * (rhog ^ Y))

Visc = Viscosity
Compress = Cg
txtVisc.Text = Viscosity
txtCompress.Text = Cg

ElseIf frmzfactor2.optGasComposition = True Then

C1 = (Val(txtC1.Text)) / 100
M1 = C1 * (16.043)
Tc1 = C1 * (-116.67 + 460#)
Pc1 = C1 * (666.4)

C2 = (Val(txtC2.Text)) / 100
M2 = C2 * (30.07)
Tc2 = C2 * (89.92 + 460#)
Pc2 = C2 * (706.5)

C3 = (Val(txtC3.Text)) / 100
M3 = C3 * (44.097)
Tc3 = C3 * (206.06 + 460#)
Pc3 = C3 * (616#)

C6 = (Val(txtC6.Text)) / 100
M6 = C6 * (86.177)
Tc6 = C6 * (453.6 + 460#)
Pc6 = C6 * (436.9)

C7 = (Val(txtC7.Text)) / 100
M7 = C7 * (100.204)
Tc7 = C7 * (512.7 + 460#)
Pc7 = C7 * (396.8)

If Val(txtC7plus.Text) > 0 Then
C7plus = (Val(txtC7plus.Text)) / 100
M7p = Val(txtM7p.Text)
M7plus = C7plus * M7p
SG7plus = Val(txtSG7plus.Text)
```

```
Tb = (4.5579 * (M7p ^ 0.15178) * (SG7plus ^ 0.15427)) ^ 3

Tc7p = 341.7 + (811 * SG7plus) + ((0.4244 + (0.1174 * SG7plus)) * Tb) + ((0.4669 - (3.2623 * SG7plus))
* (10 ^ 5) / Tb)

A = (0.24244 + (2.2898 / SG7plus) + (0.11857 / (SG7plus ^ 2))) * (0.001) * Tb
B = (1.4685 + (3.648 / SG7plus) + (0.47227 / (SG7plus ^ 2))) * (0.0000001) * (Tb ^ 2)
C = (0.42019 + (1.6977 / (SG7plus ^ 2)) * (0.0000000001) * (Tb ^ 3))
Pc7p = Exp(8.3634 - (0.0566 / SG7plus) - A + B - C)

Tc7plus = C7plus * Tc7p
Pc7plus = C7plus * Pc7p
Else

Tc7plus = 0#
Pc7plus = 0#
M7plus = 0#
End If

CO2 = (Val(txtCO2.Text)) / 100
MCO2 = CO2 * (44.01)
TcCO2 = CO2 * (87.91 + 460#)
PcCO2 = CO2 * (1071#)

H2S = (Val(txtH2S.Text)) / 100
MH2S = H2S * (34.08)
TcH2S = H2S * (212.45 + 460#)
PcH2S = H2S * (1300#)

IC4 = (Val(txtIC4.Text)) / 100
MI4 = IC4 * (58.123)
TcI4 = IC4 * (274.46 + 460#)
PcI4 = IC4 * (527.9)

IC5 = (Val(txtIC5.Text)) / 100
MI5 = IC5 * (72.15)
TcI5 = IC5 * (369.1 + 460#)
PcI5 = IC5 * (490.4)

N2 = (Val(txtN2.Text)) / 100
MN2 = N2 * (28.0134)
TcN2 = N2 * (-232.51 + 460#)
PcN2 = N2 * (493.1)

NC4 = (Val(txtNC4.Text)) / 100
MN4 = NC4 * (58.123)
TcN4 = NC4 * (305.62 + 460#)
PcN4 = NC4 * (550.6)

NC5 = (Val(txtNC5.Text)) / 100
MN5 = NC5 * (72.15)
TcN5 = NC5 * (385.8 + 460#)
PcN5 = NC5 * (488.6)

C8 = (Val(txtC8.Text)) / 100
M8 = C8 * (114.231)
Tc8 = C8 * (564.22 + 460#)
Pc8 = C8 * (360.7)
```

```
C9 = (Val(txtC9.Text)) / 100
M9 = C9 * (128.258)
Tc9 = C9 * (610.68 + 460#)
Pc9 = C9 * (331.8)

C10 = (Val(txtC10.Text)) / 100
M10 = C10 * (142.285)
Tc10 = C10 * (652# + 460#)
Pc10 = C10 * (305.2)

'Critical Properties'
Tpc = Tc1 + Tc2 + Tc3 + TcI4 + TcN4 + TcI5 + TcN5 + Tc6 + Tc7 + Tc7plus + Tc8 + Tc9 + Tc10 +
TcN2 + TcH2S + TcCO2
Ppc = Pc1 + Pc2 + Pc3 + PcI4 + PcN4 + PcI5 + PcN5 + Pc6 + Pc7 + Pc7plus + Pc8 + Pc9 + Pc10 + PcN2
+ PcH2S + PcCO2
Ma = M1 + M2 + M3 + MI4 + MN4 + MI5 + MN5 + M6 + M7 + M7plus + M8 + M9 + M10 + MN2 +
MH2S + MCO2

SG = Ma / 29

'Correction for Critical Properties'
corr = (120 * (((H2S + CO2) ^ 0.9) - ((CO2 + H2S) ^ 1.6))) + (15 * ((H2S ^ 0.5) - (H2S ^ 4#)))
Tpccorr = Tpc - corr
Ppccorr = (Ppc * Tpccorr) / (Tpc + ((H2S / 100) * (1 - (H2S / 100)) * corr))

'Reduced Properties'
  Ppr = Pseudostar / Ppccorr
  Tpr = T / Tpccorr

'Z-Factor Calculation'
If Ppr > 0.2 And Ppr < 1.2 Then
    If Tpr > 1.05 And Tpr < 1.2 Then
      z = Ppr * ((1.6643 * Tpr) - 2.2114) - (0.3647 * Tpr) + 1.4385
    ElseIf Tpr > 1.2 And Tpr < 1.4 Then
      z = Ppr * ((0.5222 * Tpr) - 0.8511) - (0.0364 * Tpr) + 1.049
    ElseIf Tpr > 1.4 And Tpr < 2# Then
      z = Ppr * ((0.1391 * Tpr) - 0.2988) + (0.0007 * Tpr) + 0.9969
    ElseIf Tpr > 2# And Tpr < 3# Then
      z = Ppr * ((0.0295 * Tpr) - 0.0825) + (0.0009 * Tpr) + 0.9967
    End If
  ElseIf Ppr > 1.2 And Ppr < 2.8 Then
    If Tpr > 1.05 And Tpr < 1.2 Then
      z = Ppr * ((-1.357 * Tpr) + 1.4942) + (4.6315 * Tpr) - 4.7009
    ElseIf Tpr > 1.2 And Tpr < 1.4 Then
      z = Ppr * ((0.1717 * Tpr) - 0.3232) + (0.5869 * Tpr) + 0.1229
    ElseIf Tpr > 1.4 And Tpr < 2# Then
      z = Ppr * ((0.0984 * Tpr) - 0.2053) + (0.0621 * Tpr) + 0.858
    ElseIf Tpr > 2# And Tpr < 3# Then
      z = Ppr * ((0.0211 * Tpr) - 0.0527) + (0.0127 * Tpr) + 0.9549
    End If
  ElseIf Ppr > 2.8 And Ppr < 5.4 Then
    If Tpr > 1.05 And Tpr < 1.2 Then
      z = Ppr * ((-0.3278 * Tpr) + 0.4752) + (1.8223 * Tpr) - 1.9036
    ElseIf Tpr > 1.2 And Tpr < 1.4 Then
      z = Ppr * ((-0.2521 * Tpr) + 0.3871) + (1.6087 * Tpr) - 1.6635
    ElseIf Tpr > 1.4 And Tpr < 2# Then
      z = Ppr * ((-0.0284 * Tpr) + 0.0625) + (0.4714 * Tpr) - 0.0011
```

```
        ElseIf Tpr > 2# And Tpr < 3# Then
            z = Ppr * ((0.0041 * Tpr) + 0.0039) + (0.0607 * Tpr) + 0.7927
        End If
    ElseIf Ppr > 5.4 And Ppr < 15# Then
        If Tpr > 1.05 And Tpr < 3# Then
            z = Ppr * ((0.711 + (3.66 * Tpr)) ^ (-1.4667)) - (1.637 / ((0.319 * Tpr) + 0.522)) + 2.071
        End If
    ElseIf Ppr > 15# Then
            z = 1
    ElseIf Ppr < 0.2 Then
            z = 1
    End If

'Compressibility'
A1 = 0.31506237
A2 = -1.0467099
A3 = -0.57832729
A4 = 0.53530771
A5 = -0.61232032
A6 = -0.10488813
A7 = 0.68157001
A8 = 0.68446549

    rho = (0.27 * Ppr) / (z * Tpr)

    D = (A1 + (A2 / Tpr) + (A3 / (Tpr ^ 3)))
    E = (2 * (A4 + (A5 / Tpr)) * rho)
    F = (5 * (A5) * (A6) * (rho ^ 4) / (Tpr))
    G = ((2 * (A7) * (rho)) / (Tpr ^ 3))
    H = (1 + (A8 * (rho ^ 2)) - ((A8 ^ 2) * (rho ^ 4)))
    I2 = (Exp((-A8) * (rho ^ 2)))

    dzdrho = D + E + F + (G * H * I2)

    Cpr = (1 / Ppr) - ((0.27 / (z ^ 2 * Tpr)) * ((dzdrho) / (1 + (rho / z) * dzdrho)))

    Cg = Cpr / Ppc

'Viscosity'

    rhog = ((2.7 * (SG) * (Pseudostar)) / ((z) * (T))) * 0.016018
    K = ((9.4 + (0.02 * Ma)) * (T ^ 1.5)) / (209# + (19# * Ma) + T)
    X = 3.5 + (986# / T) + (0.01 * Ma)
    Y = 2.4 - (0.2 * X)
    Viscosity = (K * 10 ^ (-4)) * Exp(X * (rhog ^ Y))

    Visc = Viscosity
    Compress = Cg
    txtVisc.Text = Viscosity
    txtCompress.Text = Cg
End If
End Sub

Private Sub cmdExit_Click()
End
End Sub

Private Sub txtPermeability_Change()
```

```
End Sub

Private Sub Command1_Click()

End Sub

Private Sub Text1_Change()

End Sub

Private Sub Form_Load()
txtPseudoPressure.Text = Format(Val(PseudoPress(minP)), "0")
txtPseudoPress2.Text = Format(Val(PseudoPress(minP)), "0")
txtPawf.Text = Pawf
End Sub
```

Results:

```
Private Sub cmdContinue_Click()
frmResults2.Hide
frmCalculations.Show
End Sub

Private Sub Command1_Click()
frmResults2.Hide
frmzfactor2.Show
End Sub

Private Sub Command2_Click()

End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Form_Load()
grdResults.ForeColorFixed = QBColor(4)
Dim i1 As Integer

For i1 = 1 To grdResults.Cols - 1
grdResults.ColWidth(i1) = 2000
Next i1

For j = 1 To (counter - 1)
grdResults.Col = 0
grdResults.Row = j
grdResults.Text = j
Next j

grdResults.Row = 0

grdResults.Col = 0
grdResults.CellAlignment = 1
grdResults.Text = "No_Col"

grdResults.Col = 1
grdResults.CellAlignment = 3
```

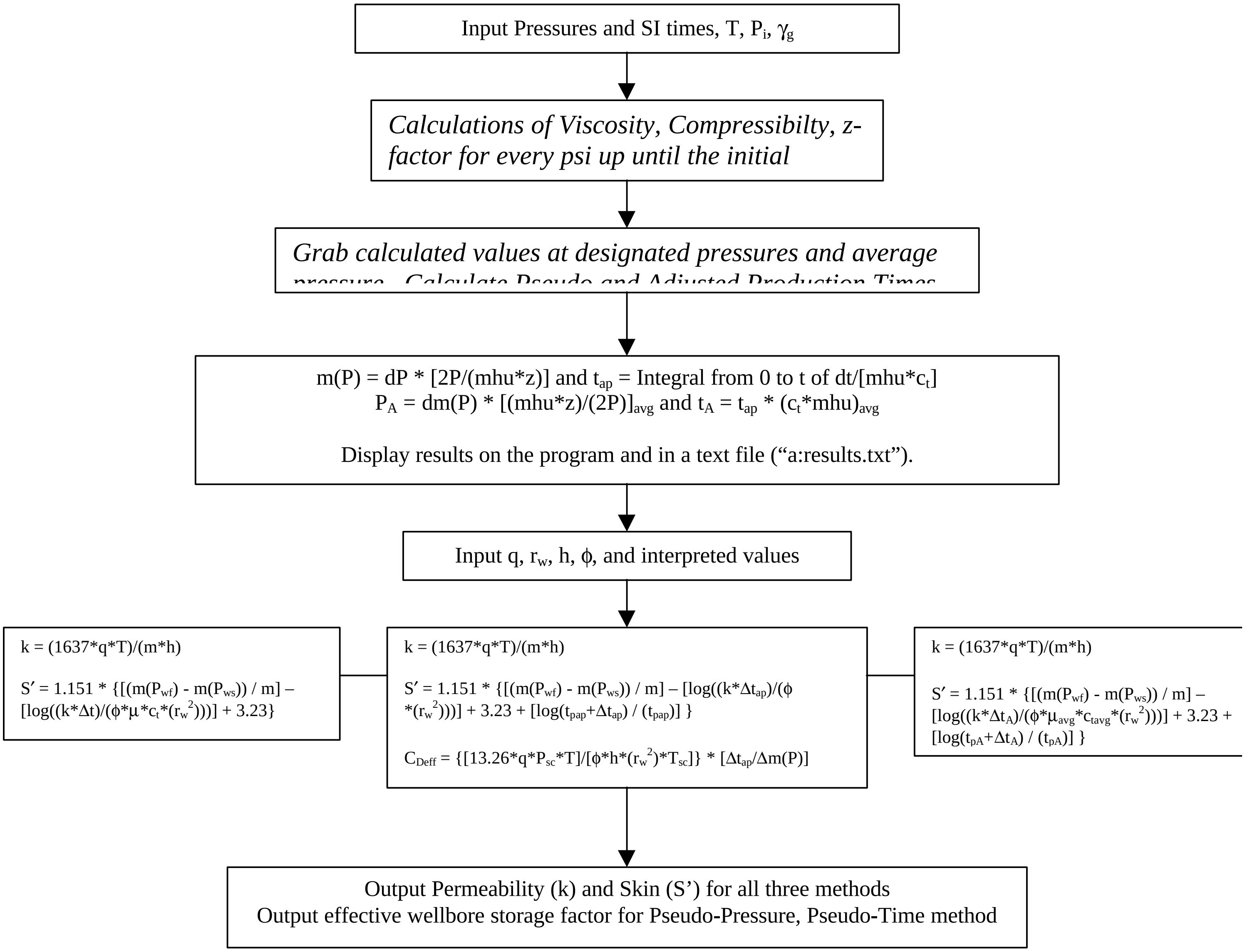
```
grdResults.Text = "Pressure"

grdResults.Col = 2
grdResults.CellAlignment = 3
grdResults.Text = "Tpr"

grdResults.Col = 3
grdResults.CellAlignment = 3
grdResults.Text = "Ppr"

grdResults.Col = 4
grdResults.CellAlignment = 3
grdResults.Text = "Z-Factor"

grdResults.Col = 5
grdResults.CellAlignment = 3
grdResults.Text = "Cg"

grdResults.Col = 6
grdResults.CellAlignment = 3
grdResults.Text = "Viscosity"

grdResults.Col = 7
grdResults.CellAlignment = 3
grdResults.Text = "2P/(mhu*z)"

grdResults.Col = 8
grdResults.CellAlignment = 3
grdResults.Text = "2P/(mhu*z))avg"

grdResults.Col = 9
grdResults.CellAlignment = 3
grdResults.Text = "Pseudo-Pressure"

grdResults.Col = 10
grdResults.CellAlignment = 3
grdResults.Text = "Adjusted Pressure"

grdResults.Col = 11
grdResults.CellAlignment = 3
grdResults.Text = "SI Time"

grdResults.Col = 12
grdResults.CellAlignment = 3
grdResults.Text = "Horner Time"

grdResults.Col = 13
grdResults.CellAlignment = 3
grdResults.Text = "Ci"

grdResults.Col = 14
grdResults.CellAlignment = 3
grdResults.Text = "Pseudo-Time"

grdResults.Col = 15
grdResults.CellAlignment = 3
grdResults.Text = "Pseudo Horner Time"
```

```
grdResults.Col = 16
grdResults.CellAlignment = 3
grdResults.Text = "Adjusted Time"

grdResults.Col = 17
grdResults.CellAlignment = 3
grdResults.Text = "Adjusted Horner Time"
End Sub

Private Sub MSFlexGrid1_Click()

End Sub
```

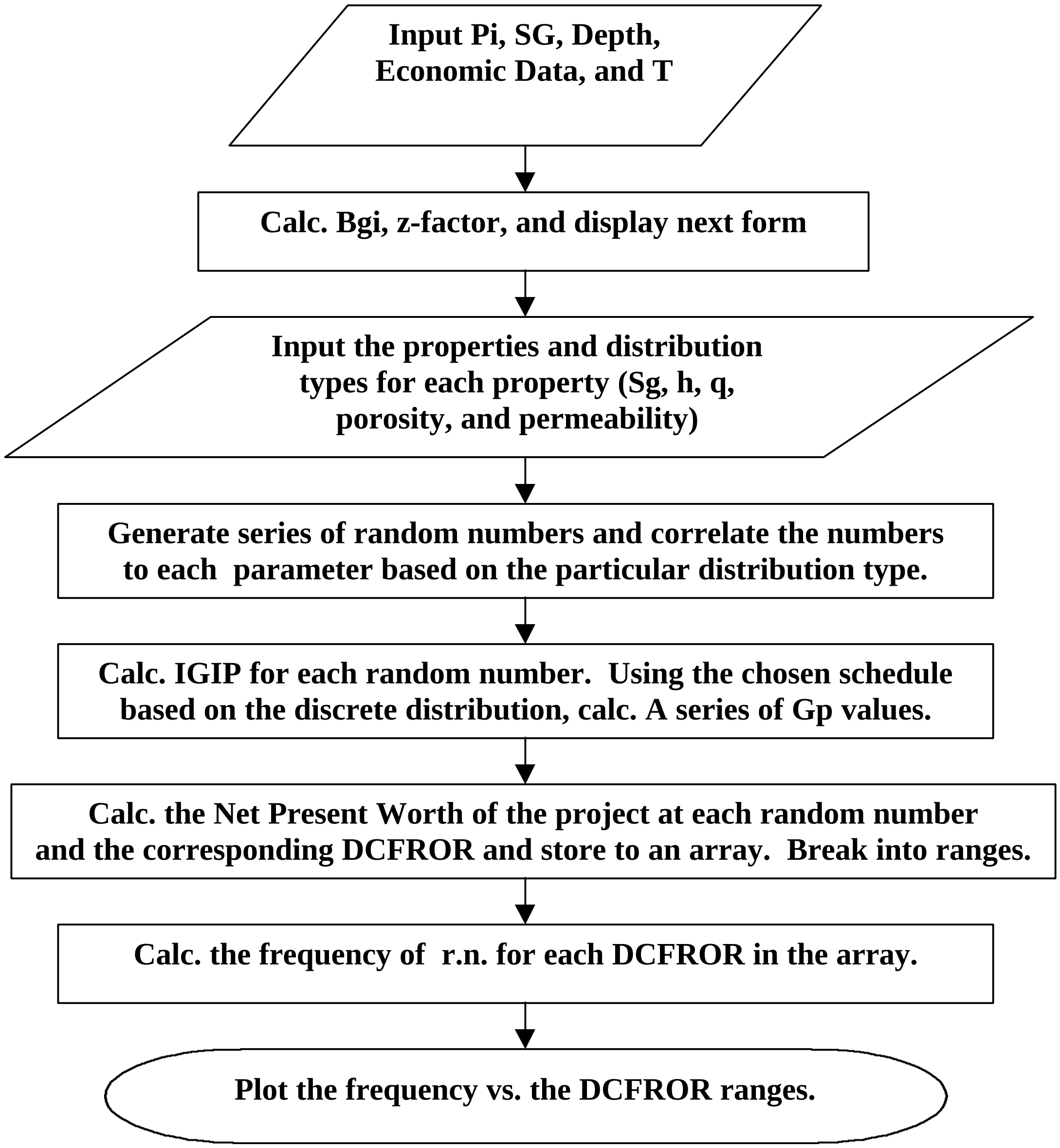
Module:

```
Public i, j, l, PAverage As Integer

Public counter, degrees_F, T, Pressure2, dPressure, Bi, PressureAdj, TimeAdj, HornerTimeAdj,
PseudoHornerTime As Double
Public Ppc, Tpc, Ppccorr, Tpccorr, corr, PT, HornerTime, PTAdj, Pinitial, PseudoPT, minP As Double
Public SG, Ma, Tpr, ViscosityAvg, CgAvg, zAvg As Double
Public A1, A2, A3, A4, A5, A6, A7, A8 As Double
Public D, K, X, Y, A, B, C As Double
Public C1, C2, C3, IC4, IC5, NC4, NC5, C6, C7, C8, C9, C10, CO2, N2, H2S As Double
Public M1, M2, M3, MI4, MI5, MN4, MN5, M6, M7, M8, M9, M10, MCO2, MN2, MH2S As Double
Public C7plus, M7p, M7plus, SG7plus, Tb, Tc7p, Pc7p, Tc7plus, Pc7plus As Double
Public PermPseudo, flow, pseudoslope, height2, SkinPseudo, PseudoPressure, PseudoPress1, Visc,
Compress, radius, adjslope, Bgavg, PermAdj, Pseudostar, SkinAdj, AdjP1, Pawf, PseudoPress2,
PseudoPressure2, pseudoslope2, PeudoTime2, PermPseudo2, SkinPseudo2, Pastar, Pstcond, Tstcond,
WBSPseudo As Double
Public dta1, dta2, dmP1, dmP2, delta As Double

Public P() As Double
Public dPmainavg() As Double
Public mainavg() As Double
Public PseudoPress() As Double
Public SITime() As Double
Public Pressure() As Double
Public main() As Double
Public Ai() As Double
Public Ci() As Double
Public Tan() As Double
Public Viscosity2() As Double
Public Cg2() As Double
Public Ppr As Double
Public rho As Double
Public E As Double
Public F As Double
Public G As Double
Public H As Double
Public I2 As Double
Public dzdrho As Double
Public Cpr As Double
Public Cg As Double
Public rhog As Double
Public z As Double
Public Viscosity As Double
```

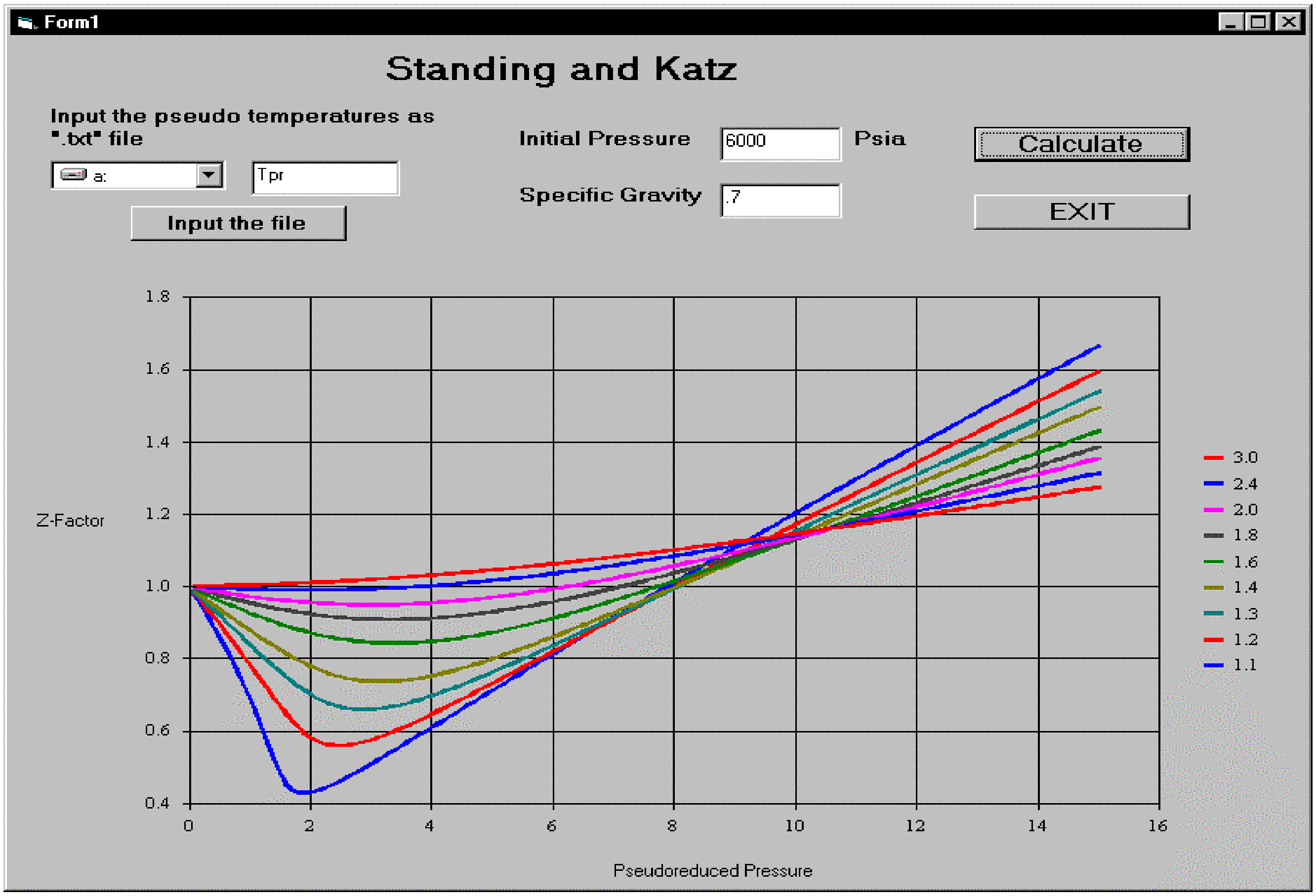
Flow Chart Permeability, Skin, Wellbore Storage:



Flow Chart for Monte Carlo Simulation:



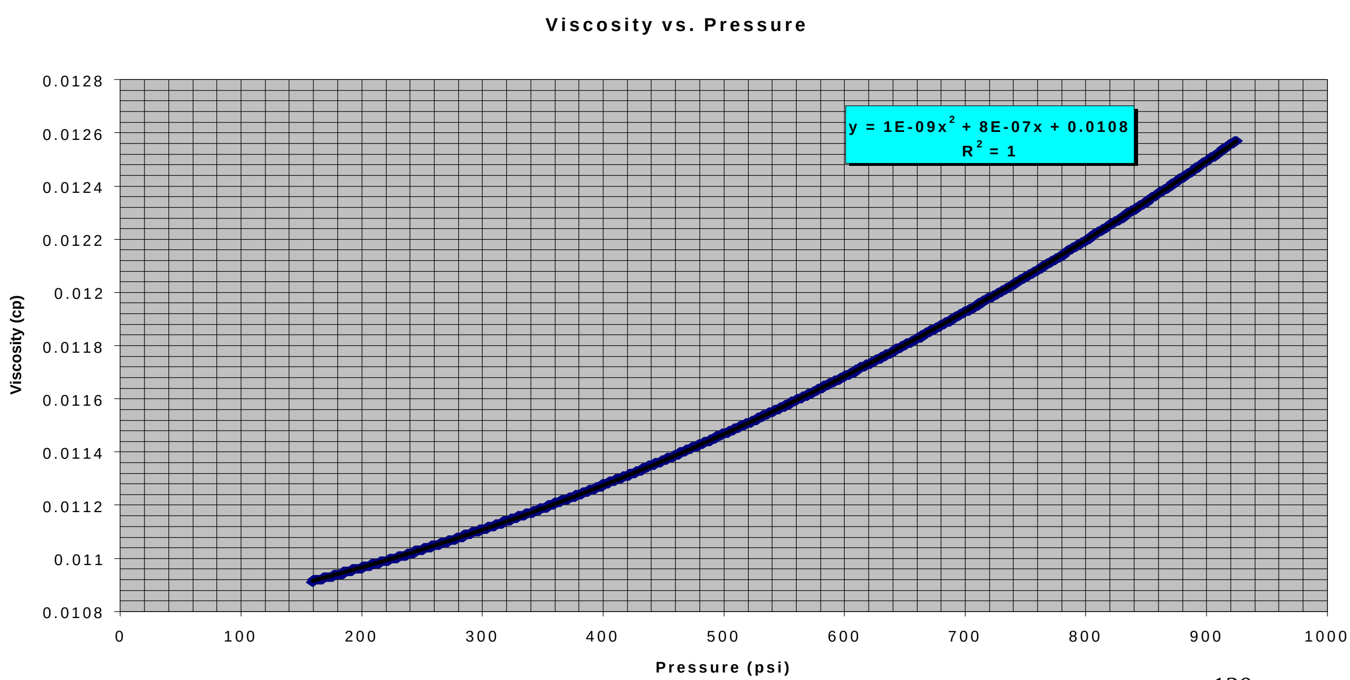
Program Recreation of Standing and Katz Relationship



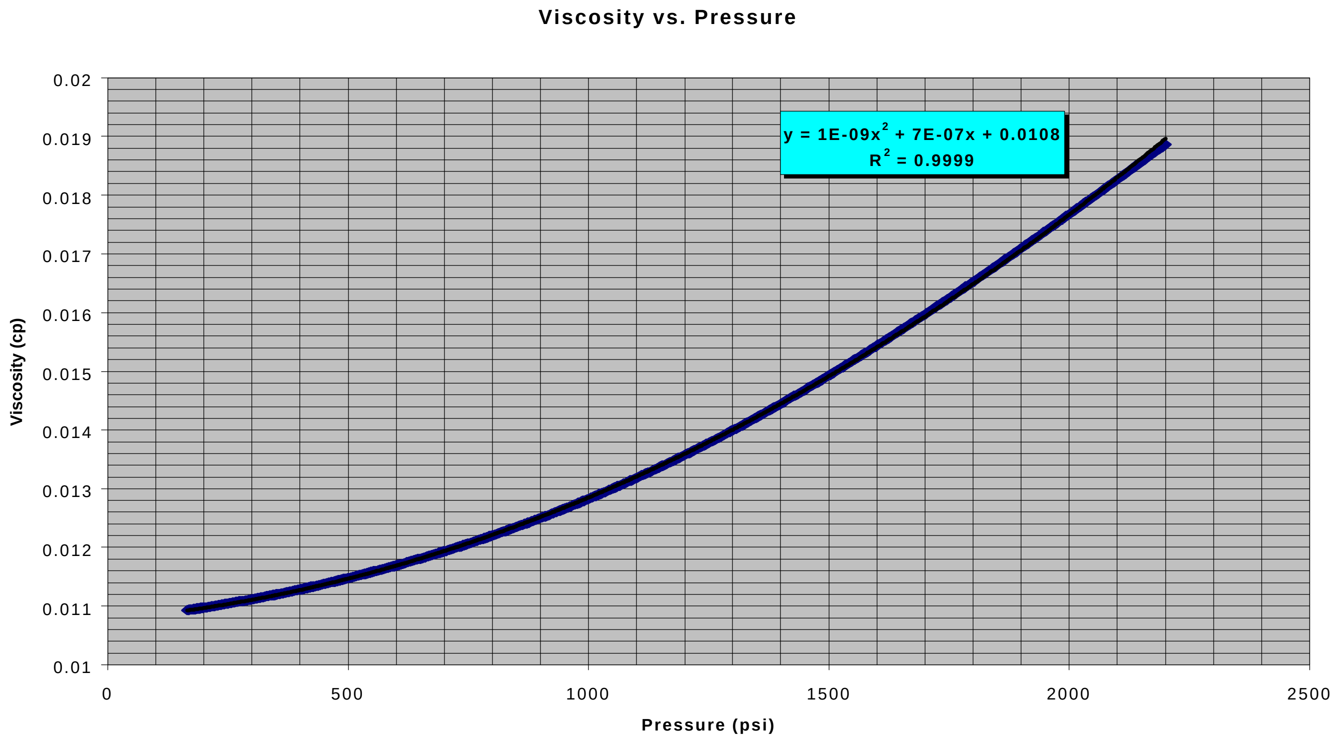
Graphs for the Viscosity vs. Pressure Relationship

Strange Creek Well:

Big Injun:

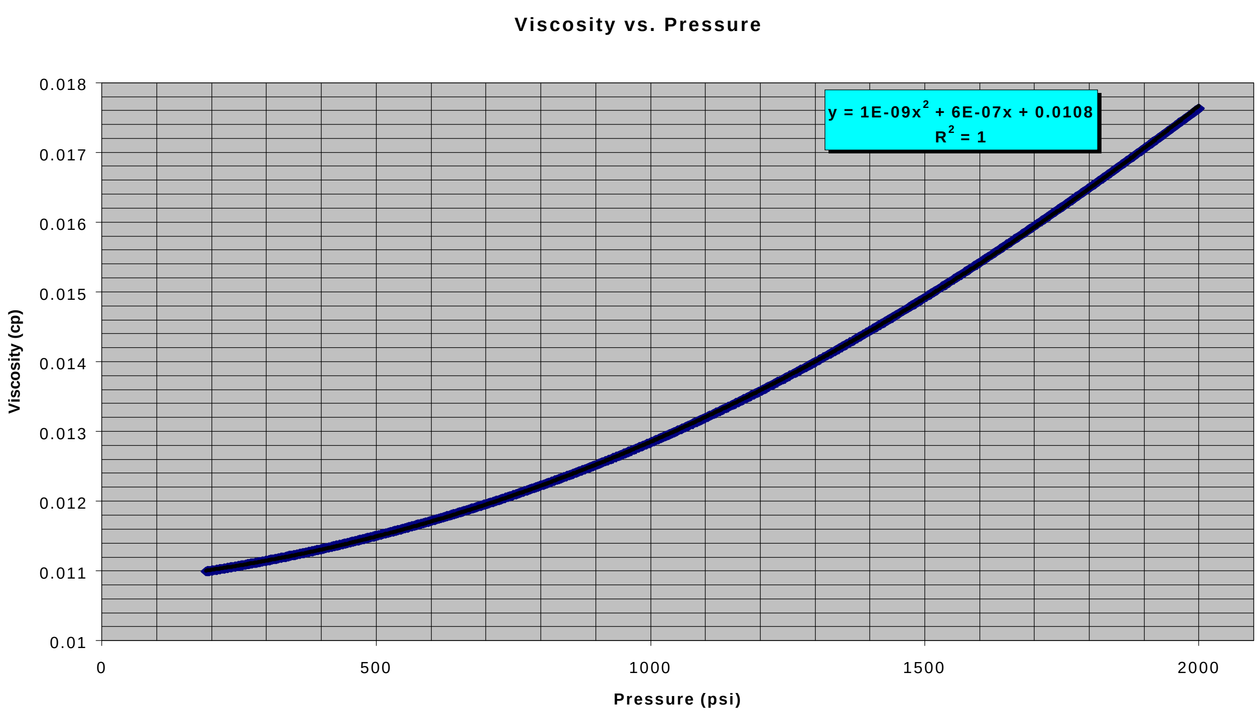


Blue Monday:



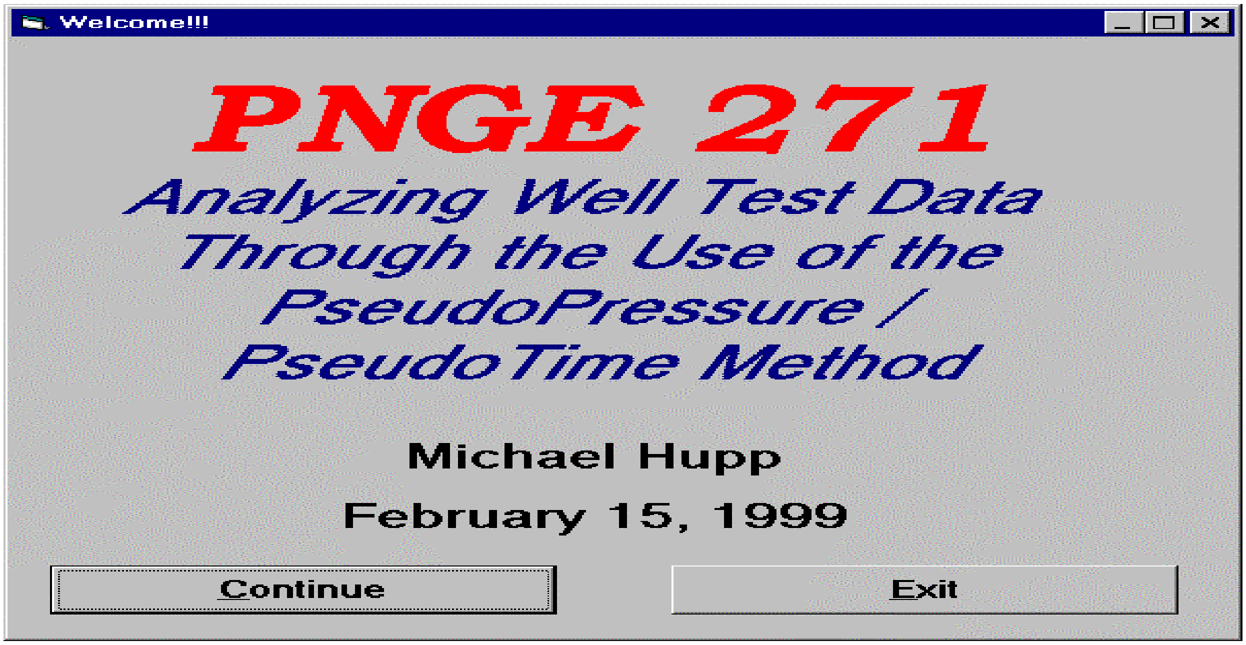
Elkhurst Well:

Weir:

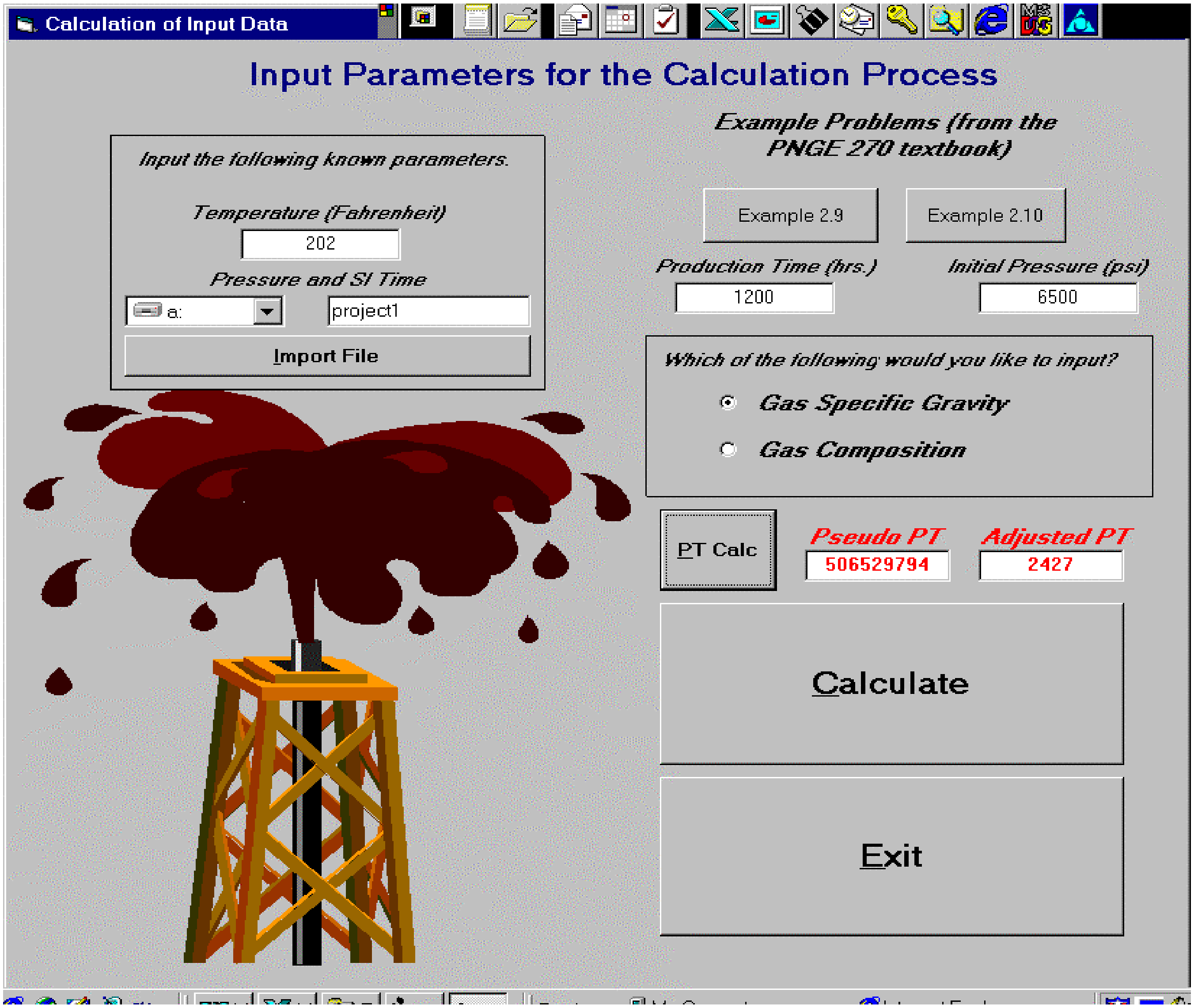


PROGRAM FORMS for PERMEABILITY and SKIN

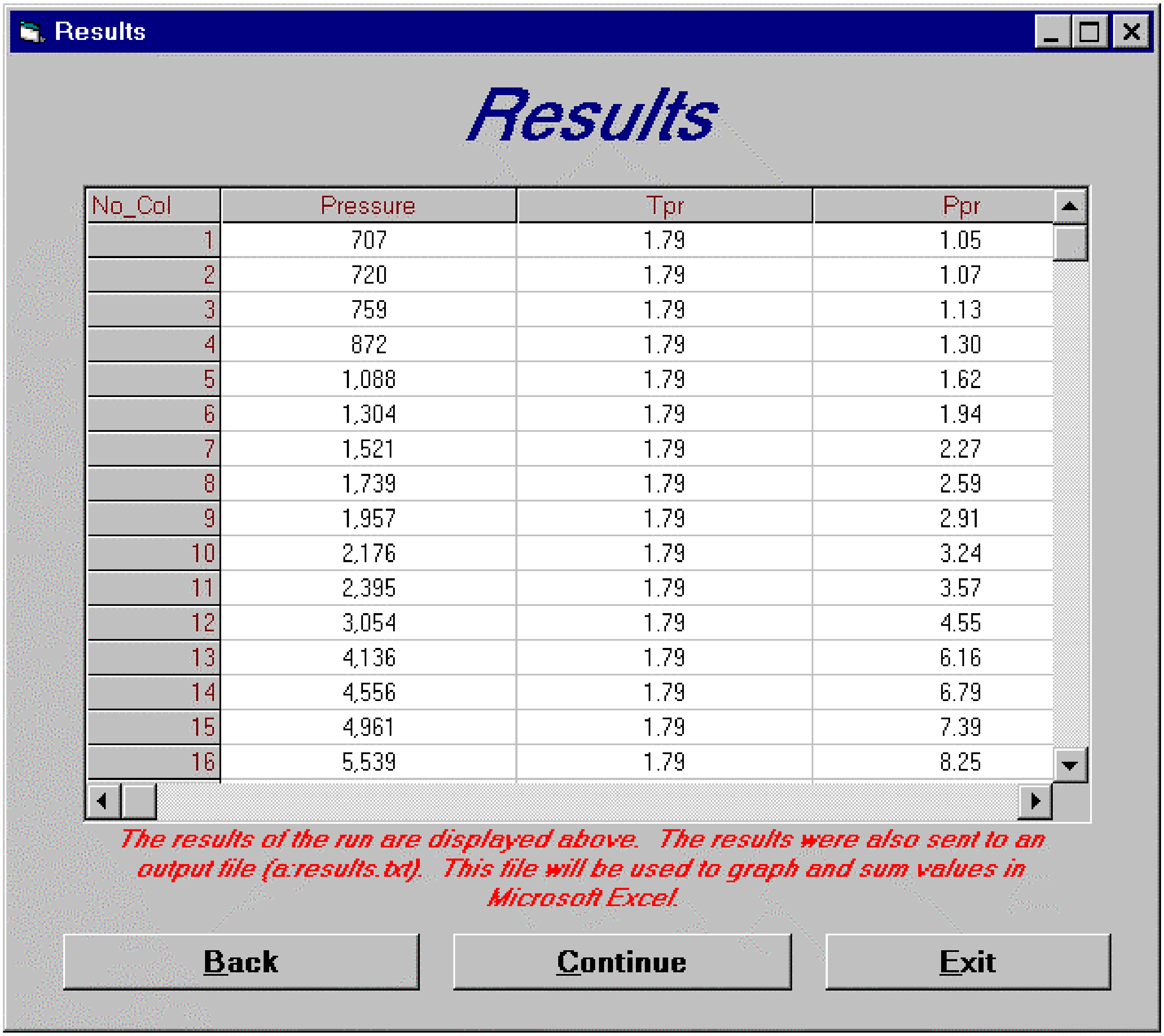
Program Forms:



The initial form for the program.



The form for calculating gas, pseudo-, and adjusted properties.


Results

Results

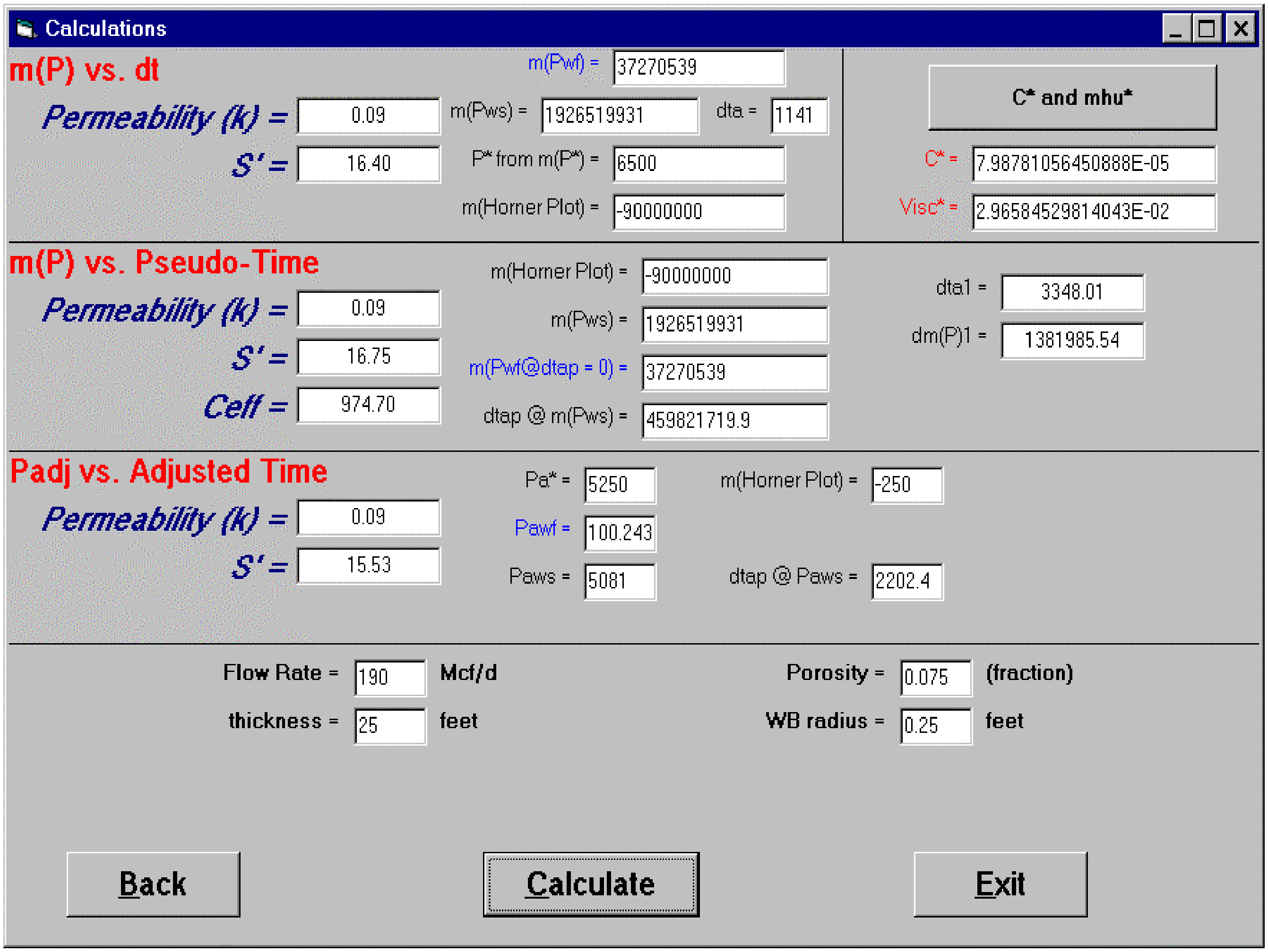
No_Col	Pressure	Tpr	Ppr
1	707	1.79	1.05
2	720	1.79	1.07
3	759	1.79	1.13
4	872	1.79	1.30
5	1,088	1.79	1.62
6	1,304	1.79	1.94
7	1,521	1.79	2.27
8	1,739	1.79	2.59
9	1,957	1.79	2.91
10	2,176	1.79	3.24
11	2,395	1.79	3.57
12	3,054	1.79	4.55
13	4,136	1.79	6.16
14	4,556	1.79	6.79
15	4,961	1.79	7.39
16	5,539	1.79	8.25

The results of the run are displayed above. The results were also sent to an output file (a:results.txt). This file will be used to graph and sum values in Microsoft Excel.

Back **Continue** **Exit**

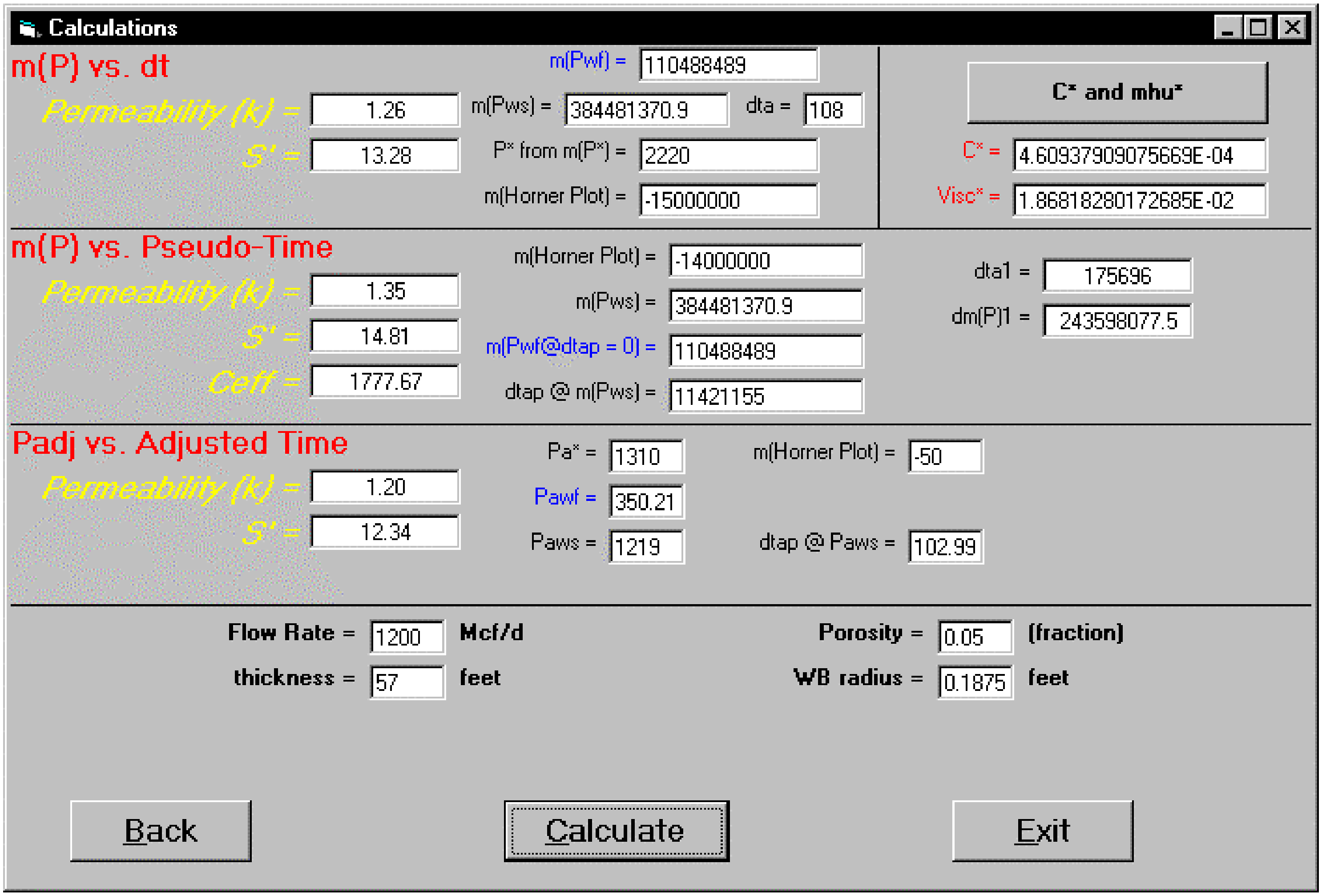
The form for displaying the results of the calculations.

PROGRAM RESULTS FORMS for PERMEABILITY and SKIN



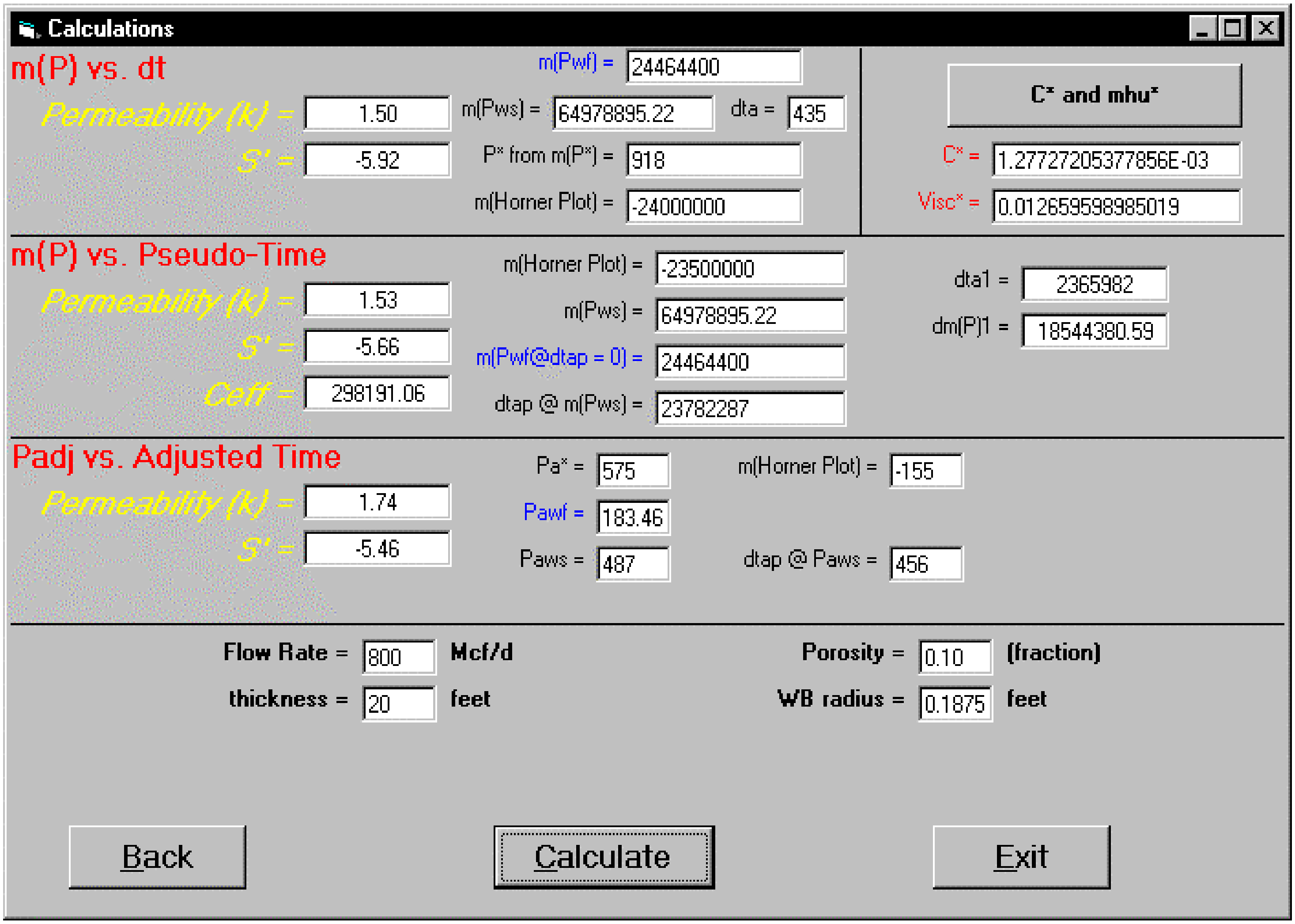
The form used to calculate the values for k, S¢, and effective wellbore storage constant (Ceff) as displayed on the form.

Big Injun:

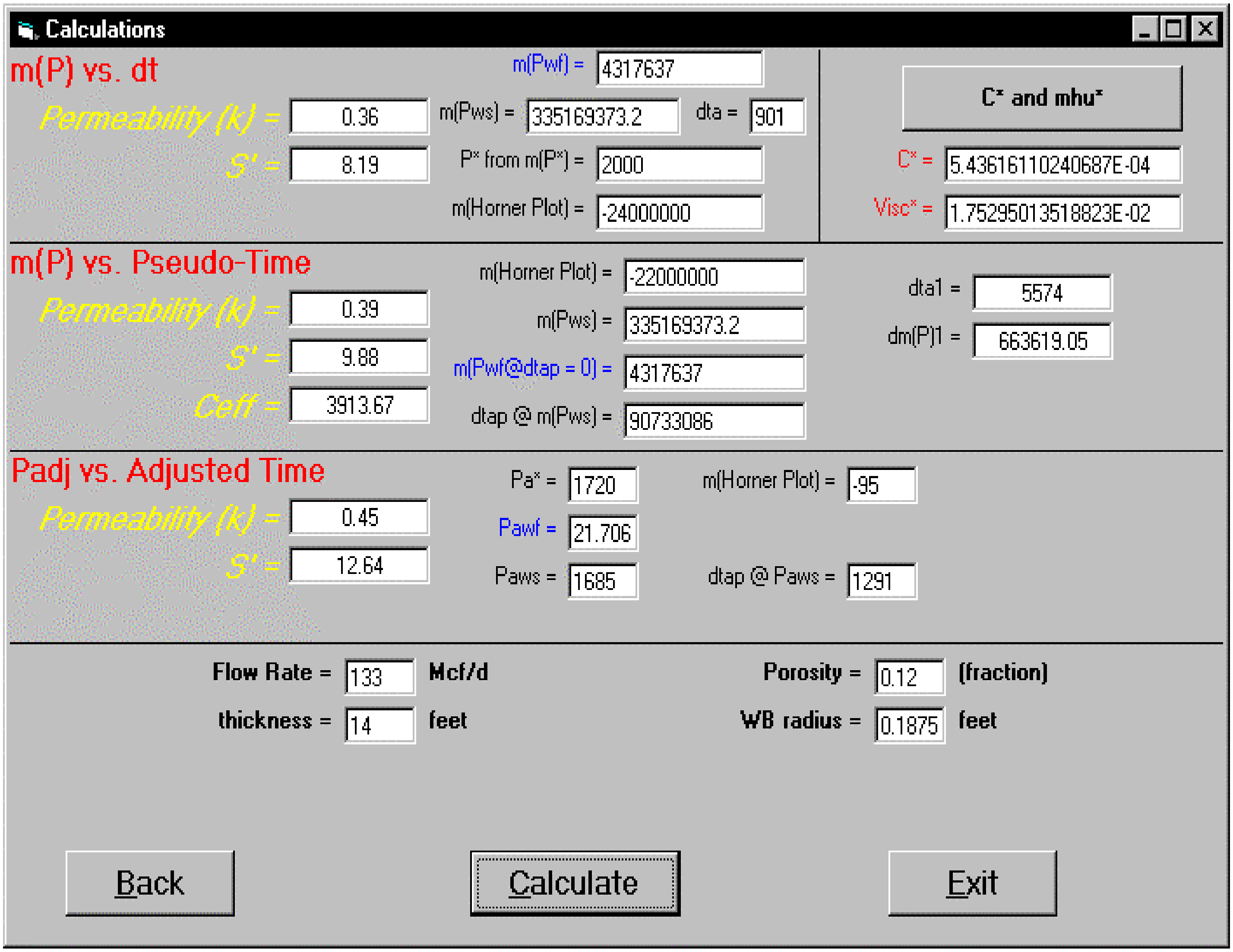


Calculations
m(P) vs. dt
Permeability (k) = 1.26
S' = 13.28
m(Pwf) = 110488489
m(Pws) = 384481370.9
dta = 108
P* from m(P*) = 2220
m(Horner Plot) = -15000000
C* and mhu*
C* = 4.60937909075669E-04
Visc* = 1.86818280172685E-02
m(P) vs. Pseudo-Time
Permeability (k) = 1.35
S' = 14.81
Ceff = 1777.67
m(Horner Plot) = -14000000
m(Pws) = 384481370.9
m(Pwf@dtap = 0) = 110488489
dtap @ m(Pws) = 11421155
dta1 = 175696
dm(P)1 = 243598077.5
Padj vs. Adjusted Time
Permeability (k) = 1.20
S' = 12.34
Pa* = 1310
Pawf = 350.21
Paws = 1219
m(Horner Plot) = -50
dtap @ Paws = 102.99
Flow Rate = 1200 Mcf/d
thickness = 57 feet
Porosity = 0.05 (fraction)
WB radius = 0.1875 feet
Back
Calculate
Exit

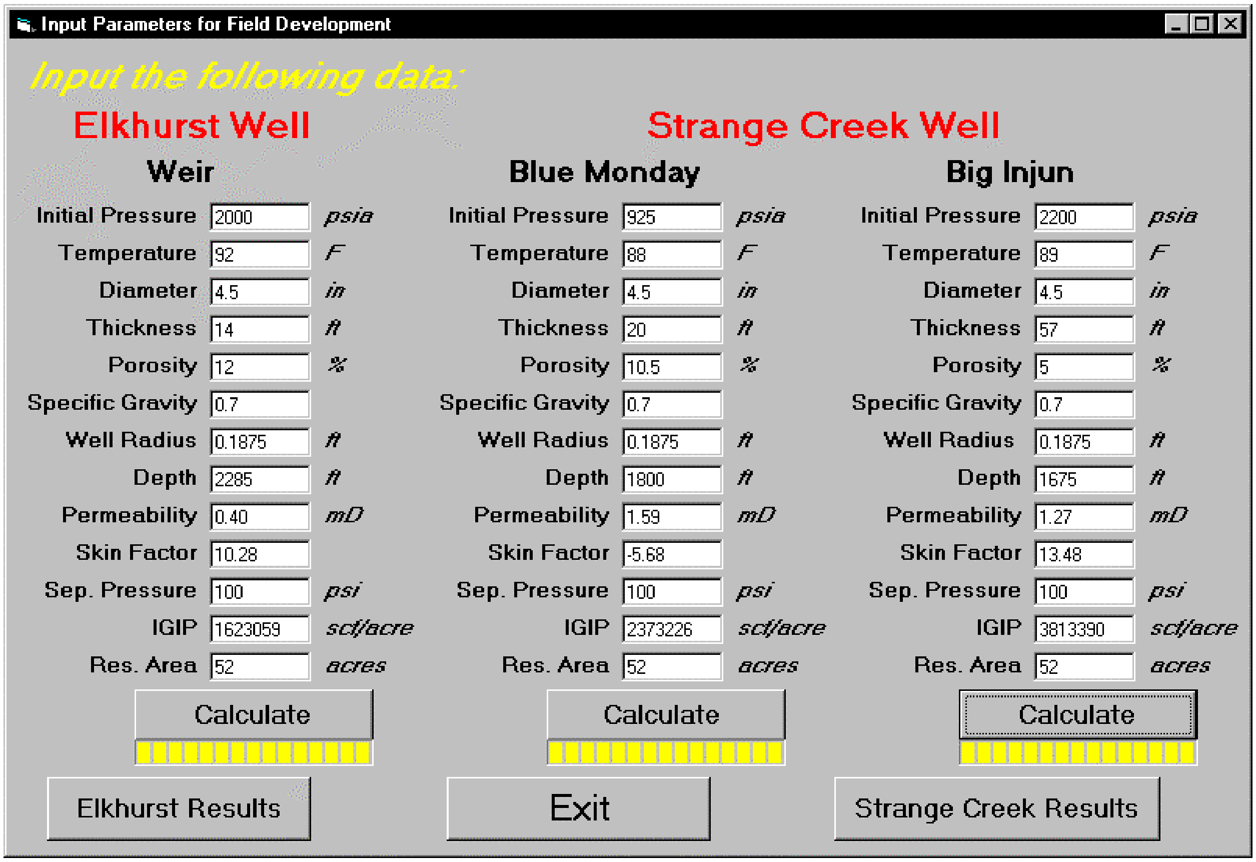
Blue Monday:

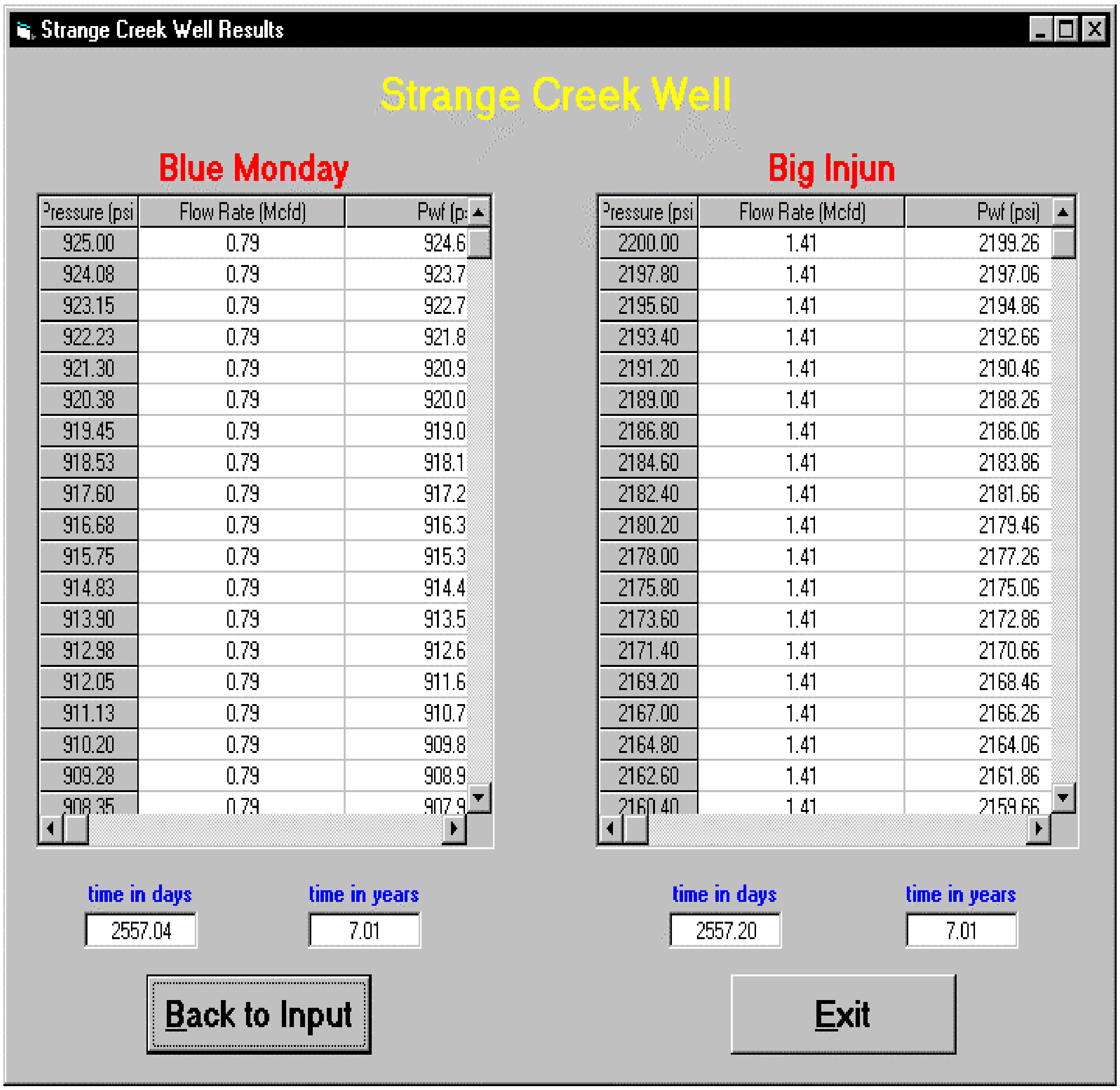

Calculations
m(P) vs. dt
m(Pwf) = 24464400
Permeability (k) = 1.50
m(Pws) = 64978895.22
dta = 435
S' = -5.92
P* from m(P*) = 918
m(Horner Plot) = -24000000
C* and mhu*
C* = 1.27727205377856E-03
Visc* = 0.012659598985019
m(P) vs. Pseudo-Time
Permeability (k) = 1.53
S' = -5.66
Ceff = 298191.06
m(Horner Plot) = -23500000
m(Pws) = 64978895.22
m(Pwf@dtap = 0) = 24464400
dtap @ m(Pws) = 23782287
dta1 = 2365982
dm(P)1 = 18544380.59
Padj vs. Adjusted Time
Permeability (k) = 1.74
S' = -5.46
Pa* = 575
Pawf = 183.46
Paws = 487
m(Horner Plot) = -155
dtap @ Paws = 456
Flow Rate = 800 Mcf/d
thickness = 20 feet
Porosity = 0.10 (fraction)
WB radius = 0.1875 feet
Back
Calculate
Exit

Weir:


Calculations
m(P) vs. dt
Permeability (k) = 0.36
S' = 8.19
m(Pwf) = 4317637
m(Pws) = 335169373.2
dta = 901
P* from m(P*) = 2000
m(Horner Plot) = -24000000
C* and mhu*
C* = 5.43616110240687E-04
Visc* = 1.75295013518823E-02
m(P) vs. Pseudo-Time
Permeability (k) = 0.39
S' = 9.88
Ceff = 3913.67
m(Horner Plot) = -22000000
m(Pws) = 335169373.2
m(Pwf@dtap = 0) = 4317637
dtap @ m(Pws) = 90733086
dta1 = 5574
dm(P)1 = 663619.05
Padj vs. Adjusted Time
Permeability (k) = 0.45
S' = 12.64
Pa* = 1720
Pawf = 21.706
Paws = 1685
m(Horner Plot) = -95
dtap @ Paws = 1291
Flow Rate = 133 Mcf/d
thickness = 14 feet
Porosity = 0.12 (fraction)
WB radius = 0.1875 feet
Back
Calculate
Exit

PROGRAM FORMS FOR OPTIMUM FLOW RATE DETERMINATION

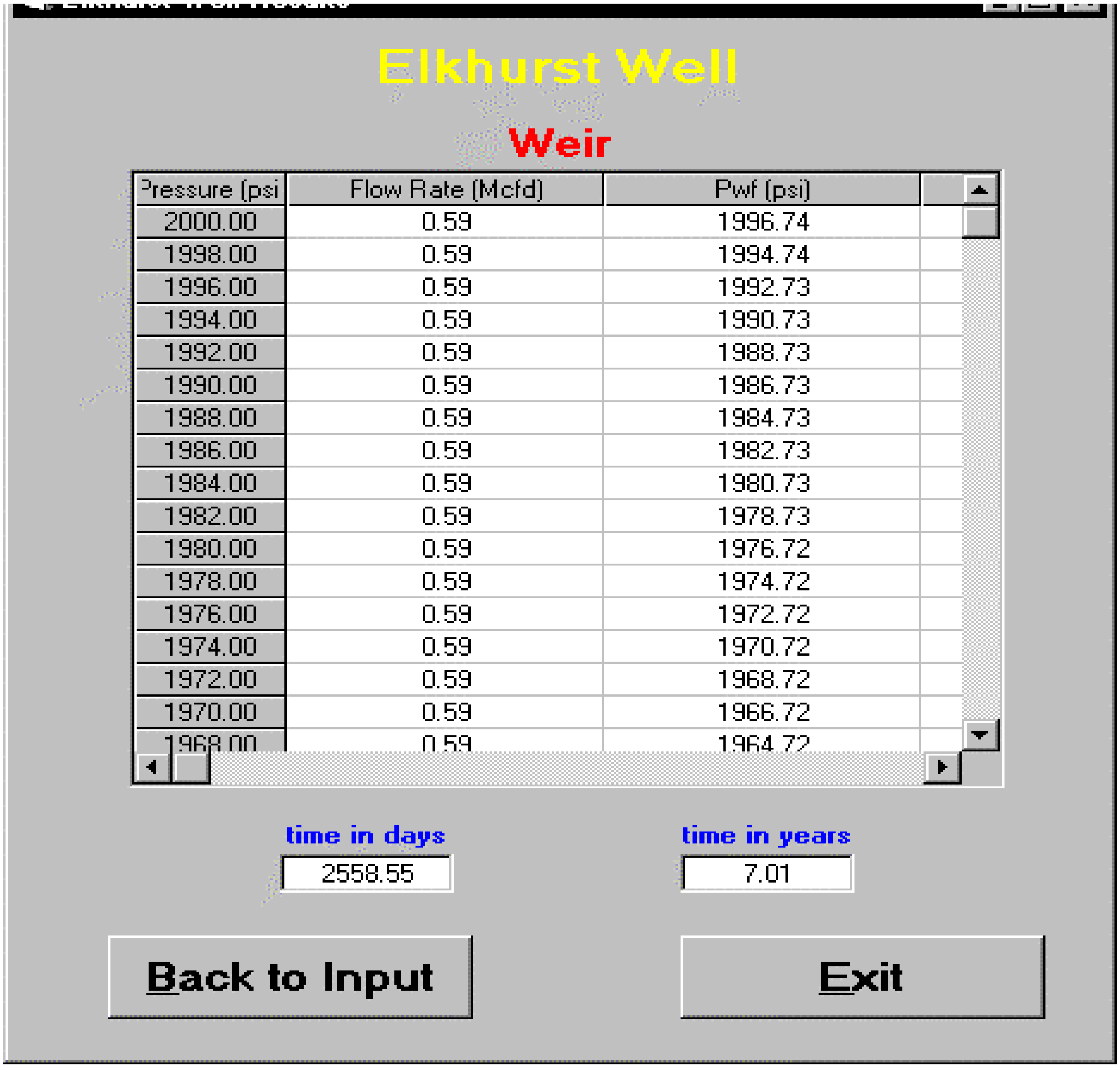




Pressure (psi	Flow Rate (Mcfd)	Pwf (p:
925.00	0.79	924.6
924.08	0.79	923.7
923.15	0.79	922.7
922.23	0.79	921.8
921.30	0.79	920.9
920.38	0.79	920.0
919.45	0.79	919.0
918.53	0.79	918.1
917.60	0.79	917.2
916.68	0.79	916.3
915.75	0.79	915.3
914.83	0.79	914.4
913.90	0.79	913.5
912.98	0.79	912.6
912.05	0.79	911.6
911.13	0.79	910.7
910.20	0.79	909.8
909.28	0.79	908.9
908.35	0.79	907.9

Pressure (psi	Flow Rate (Mcfd)	Pwf (psi)
2200.00	1.41	2199.26
2197.80	1.41	2197.06
2195.60	1.41	2194.86
2193.40	1.41	2192.66
2191.20	1.41	2190.46
2189.00	1.41	2188.26
2186.80	1.41	2186.06
2184.60	1.41	2183.86
2182.40	1.41	2181.66
2180.20	1.41	2179.46
2178.00	1.41	2177.26
2175.80	1.41	2175.06
2173.60	1.41	2172.86
2171.40	1.41	2170.66
2169.20	1.41	2168.46
2167.00	1.41	2166.26
2164.80	1.41	2164.06
2162.60	1.41	2161.86
2160.40	1.41	2159.66

:



Elkhurst Well

Weir

Pressure (psi	Flow Rate (Mcfd)	Pwf (psi)
2000.00	0.59	1996.74
1998.00	0.59	1994.74
1996.00	0.59	1992.73
1994.00	0.59	1990.73
1992.00	0.59	1988.73
1990.00	0.59	1986.73
1988.00	0.59	1984.73
1986.00	0.59	1982.73
1984.00	0.59	1980.73
1982.00	0.59	1978.73
1980.00	0.59	1976.72
1978.00	0.59	1974.72
1976.00	0.59	1972.72
1974.00	0.59	1970.72
1972.00	0.59	1968.72
1970.00	0.59	1966.72
1968.00	0.59	1964.72

time in days

2558.55

time in years

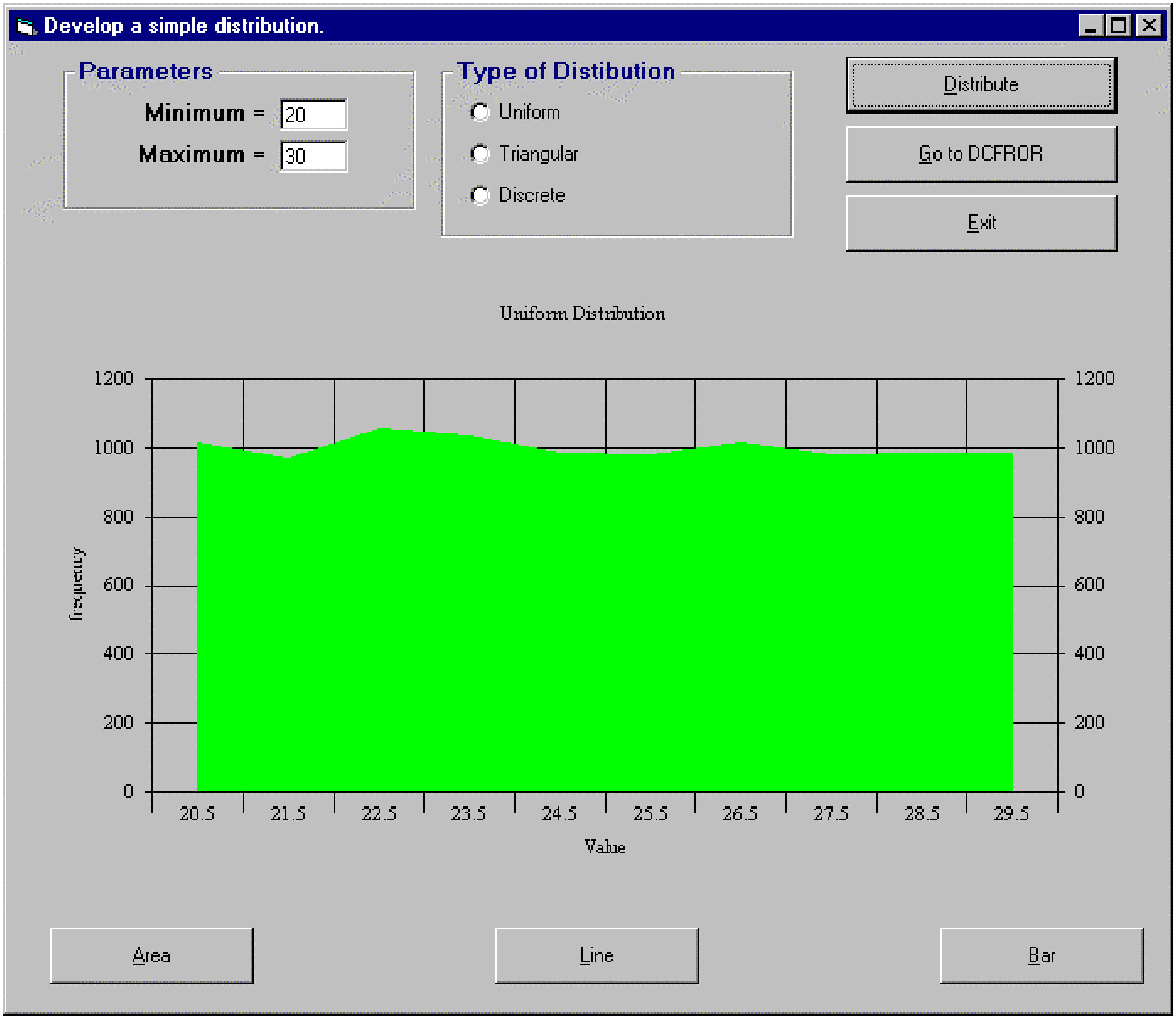
7.01

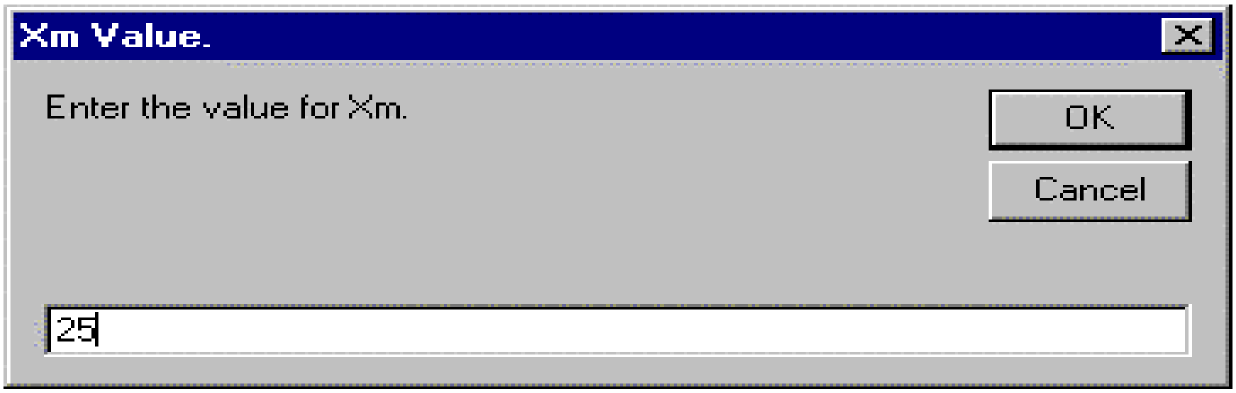
Back to Input

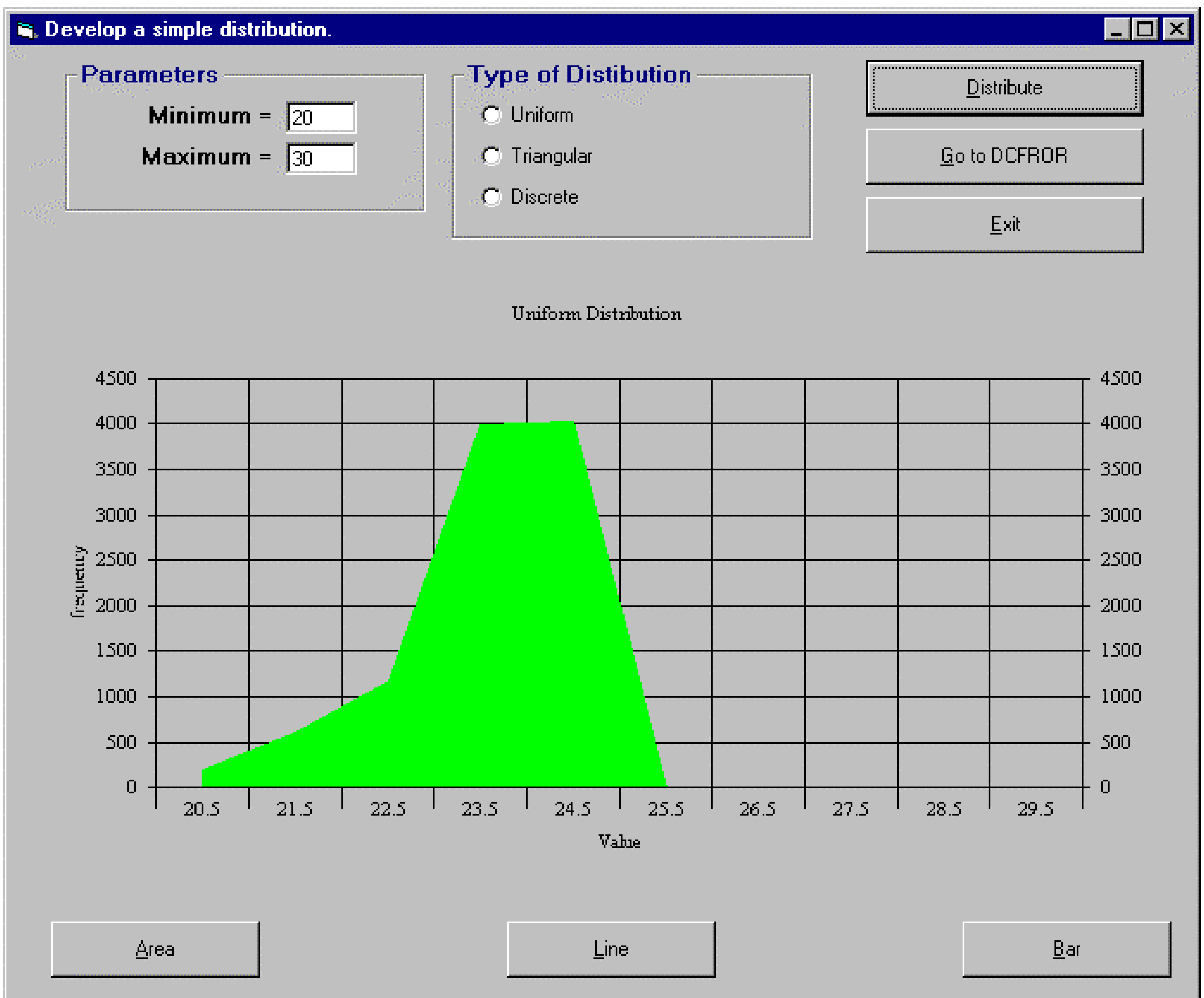
Exit

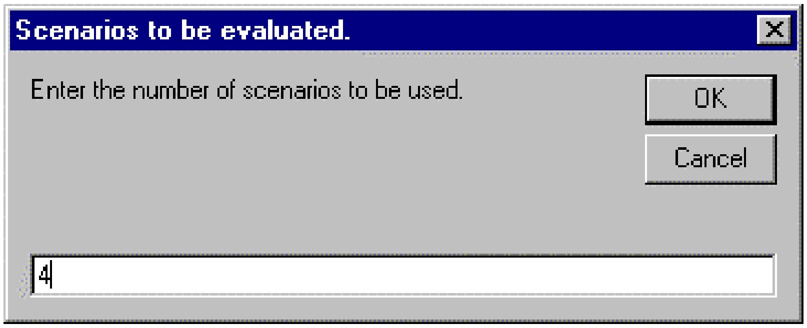
PROGRAM FORMS FOR MONTE-CARLO SIMULATION

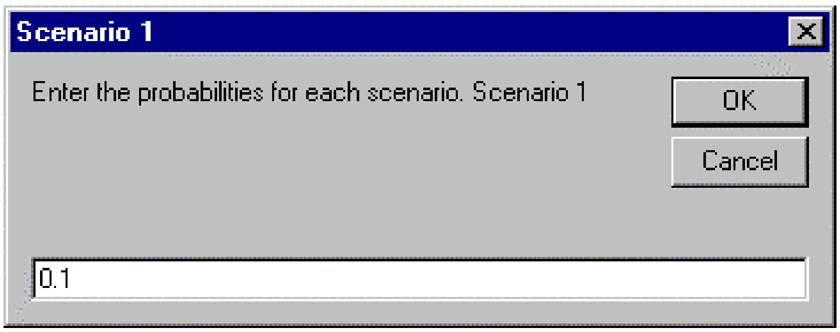
Year	1	2	3	4
1	-	-	-	-
2	-	-	-	-
3	-	-	-	-
4	0.04	0.04	0.03	0.03
5	0.17	0.13	0.12	0.1
6	0.35	0.24	0.22	0.19
7	0.56	0.42	0.35	0.29
8	0.73	0.6	0.5	0.42
9	0.85	0.73	0.62	0.53
10	0.94	0.84	0.73	0.64
11	0.99	0.91	0.82	0.73
12	1	0.96	0.89	0.82
13		0.99	0.94	0.87
14		1	0.97	0.92
15			0.99	0.95
16			1	0.98
17				0.99
18				1

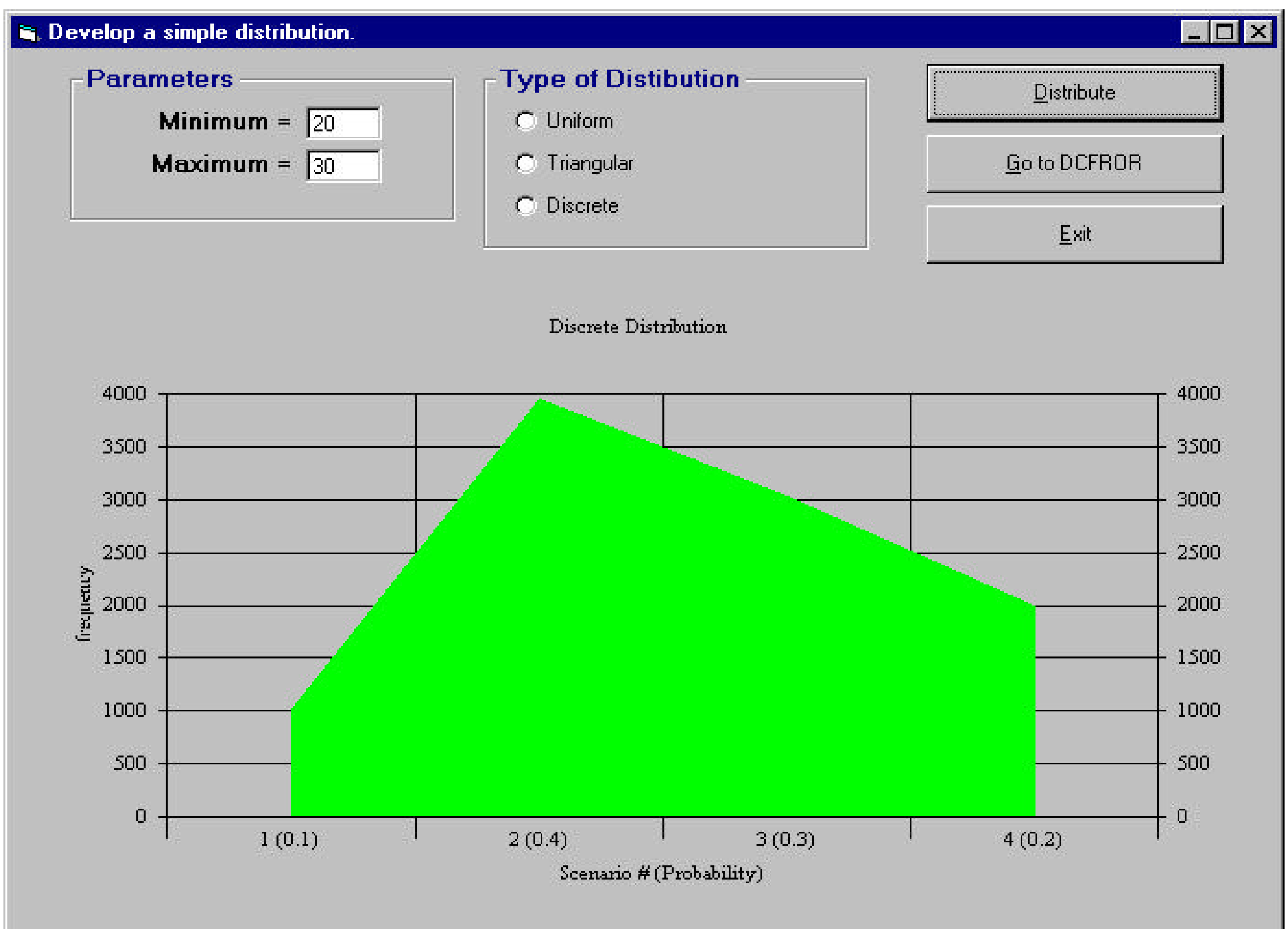

Develop a simple distribution.
Parameters
Minimum = 20
Maximum = 30
Type of Distibution
Uniform
Triangular
Discrete
Distribute
Go to DCFROR
Exit
Uniform Distribution
1200
1000
800
600
400
200
0
Frequency
20.5
21.5
22.5
23.5
24.5
25.5
26.5
27.5
28.5
29.5
Value
Area
Line
Bar


Xm Value.
Enter the value for Xm.
OK
Cancel
25


Develop a simple distribution.
Parameters
Minimum = 20
Maximum = 30
Type of Distibution
Uniform
Triangular
Discrete
Distribute
Go to DCFROR
Exit
Uniform Distribution
Frequency
Value
Area
Line
Bar

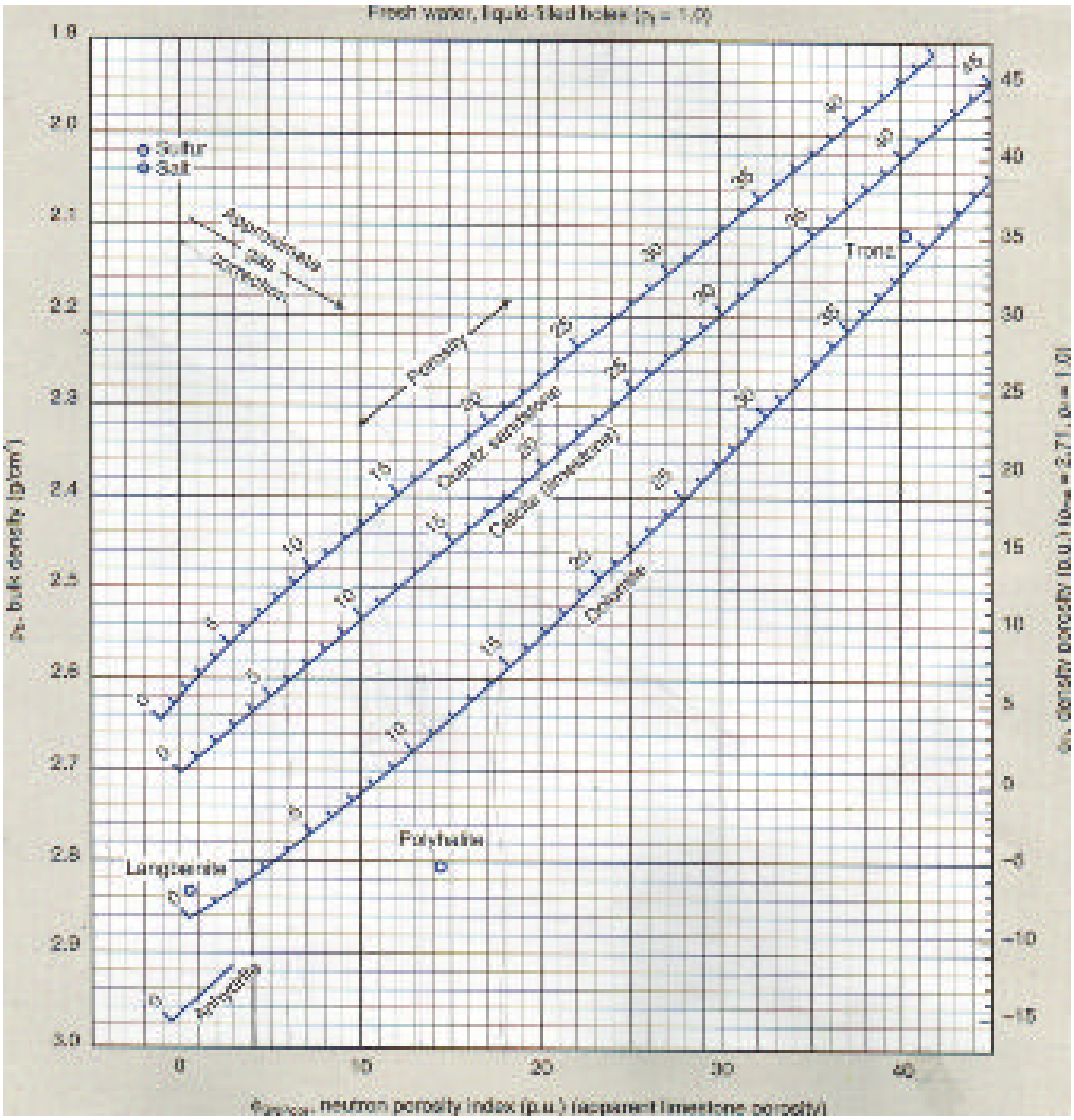

Scenarios to be evaluated.
Enter the number of scenarios to be used.
OK
Cancel
4


Scenario 1
Enter the probabilities for each scenario. Scenario 1
OK
Cancel
0.1


Develop a simple distribution.
Parameters
Minimum = 20
Maximum = 30
Type of Distibution
Uniform
Triangular
Discrete
Distribute
Go to DCFROR
Exit
Discrete Distribution
Frequency
0
500
1000
1500
2000
2500
3000
3500
4000
1 (0.1)
2 (0.4)
3 (0.3)
4 (0.2)
Scenario # (Probability)

Porosity and Lithology Determination from Formation Density Log and SNP Sidewall Neutron Porosity Log

(MAXTON FORMATION)



Crossplots for Porosity, Lithology and Gas Saturation (Sw-11)