INVESTIGATION OF ROCK STRENGTH CALCULATION USING DRILLING PARAMETERS

By

GARY JAMES BRATCHER

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1991

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1993

OKLAHOMA STATE UNIVERSITY

INVESTIGATION OF ROCK STRENGTH CALCULATION USING DRILLING PARAMETERS

Thesis Approved:

Thesis Adviser

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to thank Dr. L.L. Hoberock for his guidance and advice throughout my graduate program, and would like to express my appreciation to Dr. Richard Lowery and Dr. C. E. Price for serving on my committee. The financial support from Ercill Hunt and Associates, Houston, Texas, Amoco Production Company, Tulsa, Oklahoma, and Gas Research Institute, Chicago, Illinois is appreciated. Without this support this work would not have been possible. The advice and help of Dr. Zissis Moschovidis and Mr. Ernie Onyia, Amoco Production Company, and Ercill Hunt, Ercill Hunt and Associates, proved to be invaluable.

I would also like to thank my parents, James and Frances Lofton, for encouraging me to attend college. Most importantly, I would like to thank my wife, Teake Bratcher, for her love, support, and understanding, as well as for all of the late nights and early mornings spent organizing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION AND LITERATURE REVIEW

The work reported herein examines methods for determining in-situ rock strength in the earth, keying on work by Hareland reported in his thesis [1992], a report by Hunt, Hoberock, and Hareland to the Gas Research Institute [1992], and in a paper by Hareland and Hoberock [1993]. While the results presented here do not disagree with those presented by Hareland, this work investigates possible improvements in order to more closely determine rock strength.

Hareland developed a procedure to predict minimum principal in-situ stress using typically collected drilling data. This procedure is for use in designing hydraulic fracturing treatments for gas well stimulation in low-permeability reservoir rock called "tight gas sands". If the procedure could be reliably applied in the field, expensive fracturing stress tests would no longer be needed.

Several investigators, including Hareland and Hoberock [1993], Winters et al. [1987], Warren [1987], and Pessier and Fear [1992], have studied rock strength determination from drilling parameters. Warren [1987] developed the rate-of-penetration model used by Hareland and Hoberock [1993], and Winters et al. [1987] continued Warren's work by further developing the model to account for rock strain. While highly developed, these models require coefficients which can be difficult to obtain. Pessier and Fear [1992] elaborated on a method proposed by Teale [1965] that requires only one empirical coefficient, together with measured drilling data.

The procedure outlined by Hareland and Hoberock [1993] uses a penetration rate model for drilling with a tri-cone roller bit to determine ultimate in-situ compressive rock strength, given knowledge of drilling data. In-situ compressive rock strength is the stress at failure of the rock formation, as drilled. Compressive rock strength is a function of effective confining pressure and can be used with a plane strain assumption to obtain the Mohr failure envelope for the drilled formation. The angle of internal friction can then be determined from the Mohr failure envelope and used to calculate a "coefficient for earth at rest", denoted by K_0. Given overburden (vertical stress) and pore pressure, K_0 can be used to calculate an upper bound on the minimum horizontal stress for each foot drilled. Hareland and Hoberock [1993] achieved good results with this procedure on four experimental wells drilled by the Gas Research Institute, Chicago, Ill. These wells were named SFE (Staged Field Experiment) Nos. 1, 2, 3, and 4. The procedure has been codified by the author in a computer program, PREDICT, and is included in Appendix A.

Hareland and Hoberock emphasized that differential pressure, P_e, defined as the difference between bottom-hole bore hole pressure and pore pressure, is an important factor in both the "chip hold-down effect" and in rock strength calculations. As defined by Hareland and Hoberock [1993], the chip hold-down effect arises because the actual pressure difference across a drilled chip under the bit must be overcome before the chip can be removed by the circulating drilling mud and bit teeth. Hareland and Hoberock presented a best estimation of P_e as

$$P_e = P_B \text{, for impermeable formations} \tag{1.1}$$

and

$$P_e = P_B - P_p \text{, for permeable formations} \tag{1.2}$$

where

P_B = pressure applied to the bottom of the well bore due to the mud column weight and annular flowing friction, psi

P_p = pressure exerted on the rock formation by the rock pore fluid (pore pressure), psi

These equations ignore dynamic effects of fluid movement and also assume that all formations can be classified as either permeable or impermeable, a difficult classification to implement in practice. Assuming the true differential pressure effect occurs over approximately the same depth interval as a bit tooth penetration, dynamic influences may play a significant role in differential pressure effects. Therefore, in the study herein, it was decided to study the dynamic effects of differential pressure on drilling.

Equations (1.1 and 1.2) are oversimplified models of differential pressure. In Hareland and Hoberock's paper [1993], impermeability is defined according to Warren [1985] as "... a permeability sufficiently low that negligible pressure equalization between pores occurs over the time period in which the rock is being deformed." Given this definition for impermeable formations, all other formations were considered permeable. The question then arises, "are there better models for differential pressure?".

By using (1.1) to model differential pressure for impermeable formations, Hareland and Hoberock effectively assume that when formation overburden pressure is replaced by lower mud column pressure during drilling, the formation expands enough to reduce to zero any residual pore pressure in the formation near the bit. Equation (1.1) also assumes, by definition of impermeability, that no communication occurs between near-bit pore fluid and pore fluid remote from the bit. Accordingly, near-bit pore pressure in (1.2) is assumed to reduce to zero as a formation becomes impermeable, resulting in (1.1).

Pore pressure near the bit is believed to affect the drilling rate of penetration, R, [Warren and Smith, 1985], one of the measured variables in the penetration rate model used by Hareland and Hoberock. Warren and Smith proposed a method for determining near-bit pore pressure in impermeable formations through elastic strain theory. Warren and Smith's approach was assumed to offer advantages over Hareland and Hoberock's, which assumes near-bit pore pressure is zero, because a physical theory is provided, together with a method of determining pore pressure directly.

Equation (1.2), for differential pressure in permeable formations, assumes sufficient permeability such that pore pressure near the bore hole equals far-field, or remote, pore pressure. This equation also assumes negligible communication between drilling fluid and formation fluid, such that pressure equalization across a drilled "chip" does not occur. This assumption may introduce error because during mud cake formation on the bottom of the hole, the build-up of drilling fluid solids may have an associated "spurt loss" [Bourgoyne et al., 1986]. Spurt loss is the loss of drilling mud filtrate to the formation during initial mud cake build-up. In some cases, spurt loss could be significant enough to cause pressure equalization across a drilled chip, yielding zero differential pressure.

One complication in Hareland's [1992] method is that no means are available to handle zero differential pressure. His mathematical expression describing chip hold-down becomes indeterminate for differential pressures below 120 psi and, therefore, does not account for zero differential pressure. Any differential pressure calculated to be less than 120 psi is reset to 120 psi. This was dictated by the absence of laboratory drilling data for differential pressures below 120 psi.

As noted above, Hareland and Hoberock's approach may encounter difficulty in implementation. Considering the degree of empiricism involved with the chip hold-down and differential pressure effects, as well as the degree of uncertainty in the bit coefficients, a simpler approach might be more practical. Teale [1965] and Pessier and Fear [1992] developed an approach for determining the minimum specific energy for drilling, which is the minimum energy required to remove a unit volume of formation, and determined that this energy approximates the compressive rock strength of the given formation. The study herein further develops Pessier and Fear's approach to utilize a common drill-off test to determine rock strength.

The following work addresses each of the problems mentioned above. Chapter II develops a mathematical model of the dynamic drilling effects on differential pressure and presents results from solutions of this model. Chapter III investigates the assumptions of (1.1) and (1.2) by proposing alternate methods for determining pore and differential pressure and comparing the resulting in-situ stress bounds to Hareland and Hoberock's results. Chapter IV develops an alternate method for determining in-situ rock strength, based on the work of Teale [1965] and Pessier and Fear [1992]. Conclusions and recommendations are given in Chapter V.

CHAPTER II

DYNAMIC DRILLING EFFECTS ON DIFFERENTIAL PRESSURE FOR PERMEABLE FORMATIONS

Drilling is a dynamic process in which several events take place. Primarily, formation is removed through the combined action of bit tooth penetration and drilling mud circulation. As a result, overburden pressure on the drilled formation is removed and replaced with smaller pressures due to drilling mud. A secondary event which may occur is formation, or pore, fluid flow toward the borehole, or drilling mud or mud filtrate flow into the formation as a consequence of unequal pressures. Such flows would occur over a finite time while drilling continues. The velocity of flow would affect the dynamic pore pressure seen by the formation in the near-bit region as the hole is drilled. In turn, dynamic pore pressure affects differential pressure and, therefore, rock strength calculations. It seems appropriate to determine if dynamic effects due to fluid flow are significant enough to affect Hareland and Hoberock's [1993] static approach to differential pressure.

To assess dynamic effects on differential pressure, a one-dimensional approximation, illustrated in Figure 2.1, to a three-dimensional phenomena was developed. A one-dimensional model seems reasonable since the distance of interest ahead of the bit is typically small compared to the hole diameter. We assume that the fluid properties of the mud filtrate and formation fluid are the same. For one-dimensional flow, the continuity equation for a permeable media reduces to [Peaceman, 1977]

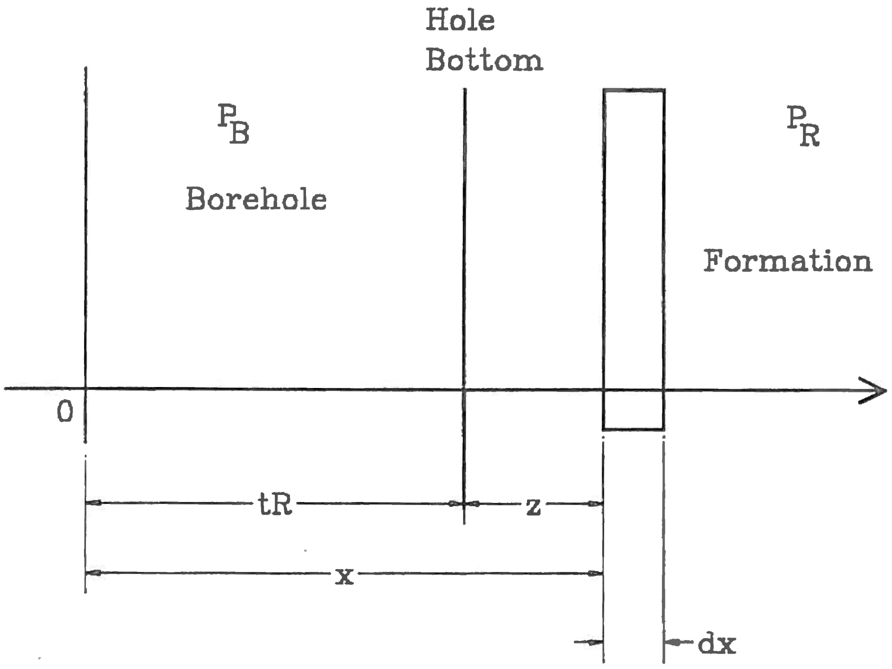


Figure 2.1 One Dimensional Dynamic Flow Model

$$\frac{\partial(\rho_1 \upsilon)}{\partial x} = -\phi \frac{\partial \rho_1}{\partial t} \tag{2.1}$$

which becomes upon expansion

$$\upsilon \frac{\partial \rho_1}{\partial x} + \rho_1 \frac{\partial \upsilon}{\partial x} = -\phi \frac{\partial \rho_1}{\partial t} \tag{2.2}$$

where

x = direction into the formation, perpendicular to the hole bottom, with origin fixed in the borehole

ρ_1 = mud filtration or formation fluid density

υ = mud filtrate or formation fluid velocity in the x-direction

ϕ = porosity of formation

The velocity of the filtrate or formation fluid is determined according to Darcy's Law [Hubbert, 1969] as

$$u = -\frac{k}{\mu}\left(\frac{\partial P_1}{\partial x}\right) \tag{2.3}$$

where

u = mud filtrate or formation fluid velocity
$P_1(x,t)$ = pressure in formation fluid or mud filtrate
k = permeability of formation
μ = absolute viscosity of formation fluid or mud filtrate

Overburden pressure is usually on the order of 1.0 psi/ft., while pressure due to the mud column may typically be on the order of 0.5 psi/ft. Many of the tight gas sand reservoirs are over 10,000 ft deep, giving a difference in pressure between overburden and mud column of 5,000 psi, or larger. Given this pressure environment, formation fluid is modeled as a compressible fluid, relating fluid density to pressure by [Nobles, 1984]

$$\rho_1 = \rho_0 e^{c(P_1 - P_0)} \tag{2.4}$$

where

P_0 = nominal pressure
ρ_0 = fluid density measured at P_0
c = compressibility of the fluid, assumed constant

The appropriate derivatives in (2.2) are

$$\frac{\partial \rho_1}{\partial x} = \rho_0 c e^{c(P_1 - P_0)} \frac{\partial P_1}{\partial x} \tag{2.5}$$

$$\frac{\partial \rho_1}{\partial t} = \rho_0 c e^{c(P_1 - P_0)} \frac{\partial P_1}{\partial t} \tag{2.6}$$

and

$$\frac{\partial u}{\partial x} = \frac{\partial}{\partial x}\left(-\frac{k}{\mu}\left(\frac{\partial P_1}{\partial x}\right)\right) \tag{2.7}$$

Upon substitution, (2.2) becomes

$$-\left(\frac{k}{\mu}\frac{\partial P_1}{\partial x}\right)\left(\rho_0 c e^{c(P_1 - P_0)}\frac{\partial P_1}{\partial x}\right) + \left(\rho_0 e^{c(P_1 - P_0)}\right)\left(\frac{\partial}{\partial x}\left(-\left(\frac{k}{\mu}\frac{\partial P_1}{\partial x}\right)\right)\right) = -\phi \rho_0 c e^{c(P_1 - P_0)}\frac{\partial P_1}{\partial t} \tag{2.8}$$

Now, combine terms and assume k and μ are constant to obtain

$$\frac{kc}{\mu}\left(\frac{\partial P_1}{\partial x}\right)^2 + \frac{k}{\mu}\left(\frac{\partial^2 P_1}{\partial x^2}\right) = \phi c \frac{\partial P_1}{\partial t} \tag{2.9}$$

Because c is quite small, the first term on the left of (2.9) is negligibly small compared to the second. Hence, (2.9) may be reduced to

$$\frac{k}{\mu}\frac{\partial^2 P_1}{\partial x^2} = \phi c \frac{\partial P_1}{\partial t} \tag{2.10}$$

Now let us recast this problem as a moving boundary problem. Let z represent distance into the formation, with the origin at the moving hole bottom, such that

$$z = x - Rt \tag{2.11}$$

where R is the rate-of-penetration, assumed constant. Then, for (2.10), we have

$$\frac{k}{\mu}\frac{\partial}{\partial z}\left(\frac{\partial P}{\partial z}\frac{\partial z}{\partial x}\right)\frac{\partial z}{\partial x} = \phi c\left(\frac{\partial P}{\partial z}\frac{\partial z}{\partial t} + \frac{\partial P}{\partial t}\right) \tag{2.12}$$

where

$$P_1(x,t) = P(z,t)$$

Using (2.11) the appropriate derivatives in (2.12) are

$$\frac{\partial z}{\partial x} = 1 \tag{2.13}$$

$$\frac{\partial z}{\partial t} = -R \tag{2.14}$$

Substitution in (2.12) yields

$$\frac{k}{\mu}\frac{\partial^2 P}{\partial z^2} = \phi c\left(\frac{\partial P}{\partial t} - R\frac{\partial P}{\partial z}\right) \tag{2.15}$$

Rearranging yields

$$\frac{\partial^2 P}{\partial z^2} + \frac{R}{\alpha}\frac{\partial P}{\partial z} = \frac{1}{\alpha}\frac{\partial P}{\partial t} \tag{2.16}$$

where

$$\alpha = \frac{k}{\phi c \mu} \tag{2.17}$$

For time approaching infinity the steady-state equation for (2.16) may be determined by setting

$$\frac{\partial P}{\partial t} = 0, \; t \Rightarrow \infty \tag{2.18}$$

which then gives

$$\frac{d^2 P}{dz^2} + \frac{R}{\alpha}\frac{dP}{dz} = 0 \tag{2.19}$$

Integrating twice yields

$$P = c_2 + c_1 e^{-\left(\frac{R}{\alpha}z\right)} \tag{2.20}$$

In order to evaluate the two integration constants, c_1 and c_2, the boundary conditions must be specified. At the bottom of the hole, z = 0, the pressure is the bottom hole mud pressure, such that $P(0) = P_B$. We take as a second "boundary" a multiple n of the bit diameter, D, at which we set the pressure equal to the in-situ pore pressure, P_R, such that $P(nD) = P_R$. Then (2.20) becomes

$$P = \frac{P_B\left(e^{-\left(\frac{R}{\alpha}z\right)} - e^{-\left(\frac{RnD}{\alpha}\right)}\right) - P_R\left(e^{-\left(\frac{R}{\alpha}z\right)} - 1\right)}{1 - e^{-\left(\frac{RnD}{\alpha}\right)}} \tag{2.21}$$

where

P_B = bottom hole pressure due to mud column weight and frictional pressure
P_R = pore pressure remote from the bore hole
n = multiple of the bit diameter
D = bit diameter

The differential pressure drop through distance z ahead of the bottom hole is then given by

$$\Delta(z) = P_B - P(z) \tag{2.22}$$

We are primarily interested in the pressure drop across the thickness, h, of a rock chip because this is the distance of concern in rock strength calculations. While chip thicknesses vary considerably, for the results that follow we select a value of h = 0.10 inches as typical. Then, setting z = h and using (2.21) in (2.22) gives

$$\Delta(h) = (P_B - P_R)\frac{1-e^{-\left(\frac{R}{\alpha}h\right)}}{1-e^{-\left(\frac{RnD}{\alpha}\right)}} \tag{2.23}$$

In order to use units in (2.23) that are commonly used in practice, we introduce a unit conversion constant, ψ, which for (2.23), yields

$$\frac{\Delta(h)}{(P_B - P_R)} = \frac{1-e^{-\left(\frac{\psi R}{\alpha}h\right)}}{1-e^{-\left(\frac{\psi RnD}{\alpha}\right)}} \tag{2.24}$$

where

$\Delta(h)$ = change in pressure over depth h, psi
R = rate-of-penetration, ft/hr
P_R = pore pressure remote from the borehole, psi
nD = multiple of bit diameter, inches
k = permeability, md
μ = absolute viscosity of formation fluid, cp
ϕ = porosity, decimal fraction
c = compressibility, 1/psi
ψ = conversion factor, 316 (lbf md hr)/(ft in^3 cp)
α = $k/(\mu\phi c)$, md psi / cp

Observe that, as nD approaches infinity, (2.24) becomes

$$\frac{\Delta(h)}{P_B - P_R} = \left(1-e^{-\left(\frac{\psi R}{\alpha}h\right)}\right), \quad nD \Rightarrow \infty \tag{2.25}$$

Also, note that in the case where no drilling occurs, R = 0, (2.24) reduces to the static pressure drop case, namely

$$\frac{\Delta(h)}{P_B - P_R} = \frac{h}{nD}, \qquad R=0 \qquad (2.26)$$

A computer program, DYNAMIC, was written to solve (2.24), such that the sensitivity of (2.24) to variations in important parameters could be determined. Nominal values for ϕ, c, and μ were set at 0.10, 0.0001 1/psi, and 1.0 cp, respectively, which represent a typical tight gas sand formation, with water as the formation fluid. Permeability was allowed to range from one microdarcy to one millidarcy, again typical values for a tight gas sand formation. The value of nD is somewhat arbitrary, so nD was allowed to range over several orders of magnitude to determine the effect on Δ(h), and in the limit nD approaches infinity. The computer program DYNAMIC is given in Appendix B, while the results are shown in Figures 2.2-2.4.

Figures 2.2-2.4 present results for Δ(h) over a range of typical penetration rates used in drilling tight gas sand formations, with 40 ft/hr representing an exceptionally high maximum for most deep hole drilling. The results illustrate that dynamic differential pressure varies over this range and approaches Hareland and Hoberock's [1993] static, or maximum, differential pressure with increasing penetration rate. Also apparent from the figures is the decreasing effect of nD as nD increases. In fact, for values of n larger than 5, the effect of increasing nD is negligible.

The effect of permeability on dynamic differential pressure can be more easily understood by defining a new term, υ_{fh}, as the formation fluid velocity at a distance h. Rewrite (2.25) as

$$\frac{\Delta(h)}{P_B - P_R} = \left(1 - e^{-\left(\frac{\upsilon_b}{\upsilon_{fh}}\right)} \right) \qquad (2.27)$$

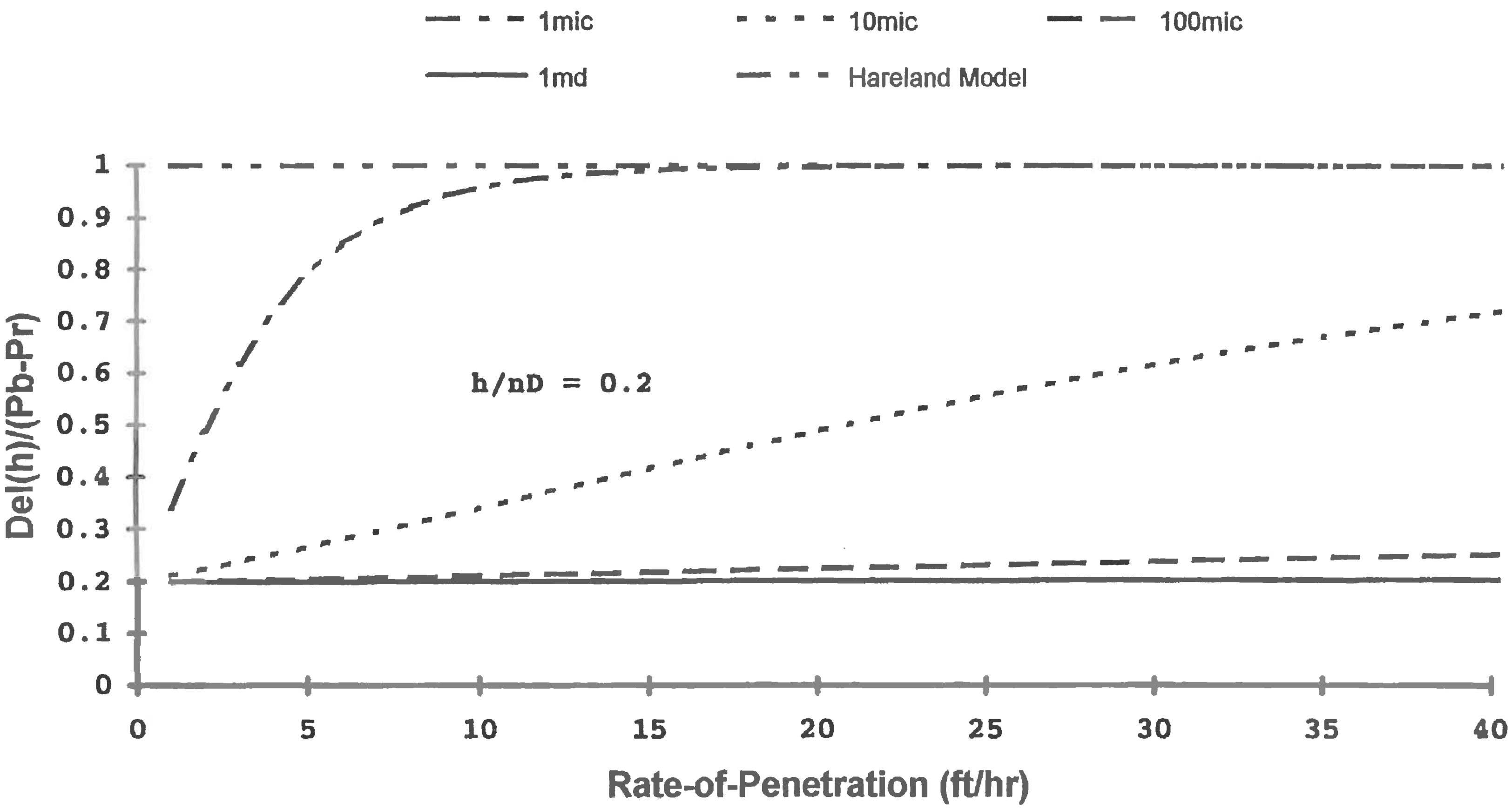


Figure 2.2 Dynamic Differential Pressure Study - nD = 0.5 in.

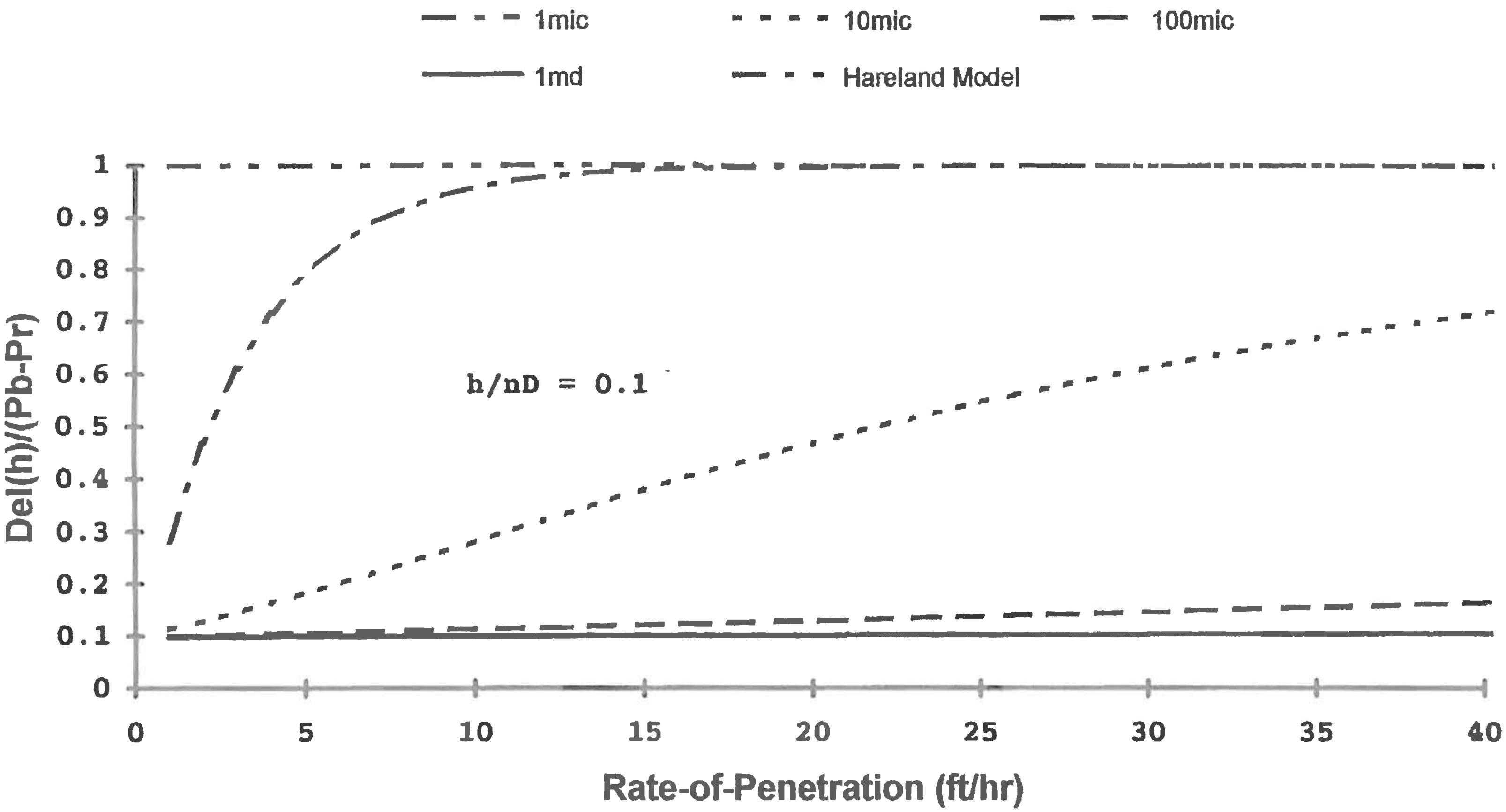


Figure 2.3 Dynamic Differential Pressure Study - nD = 1.0 in.

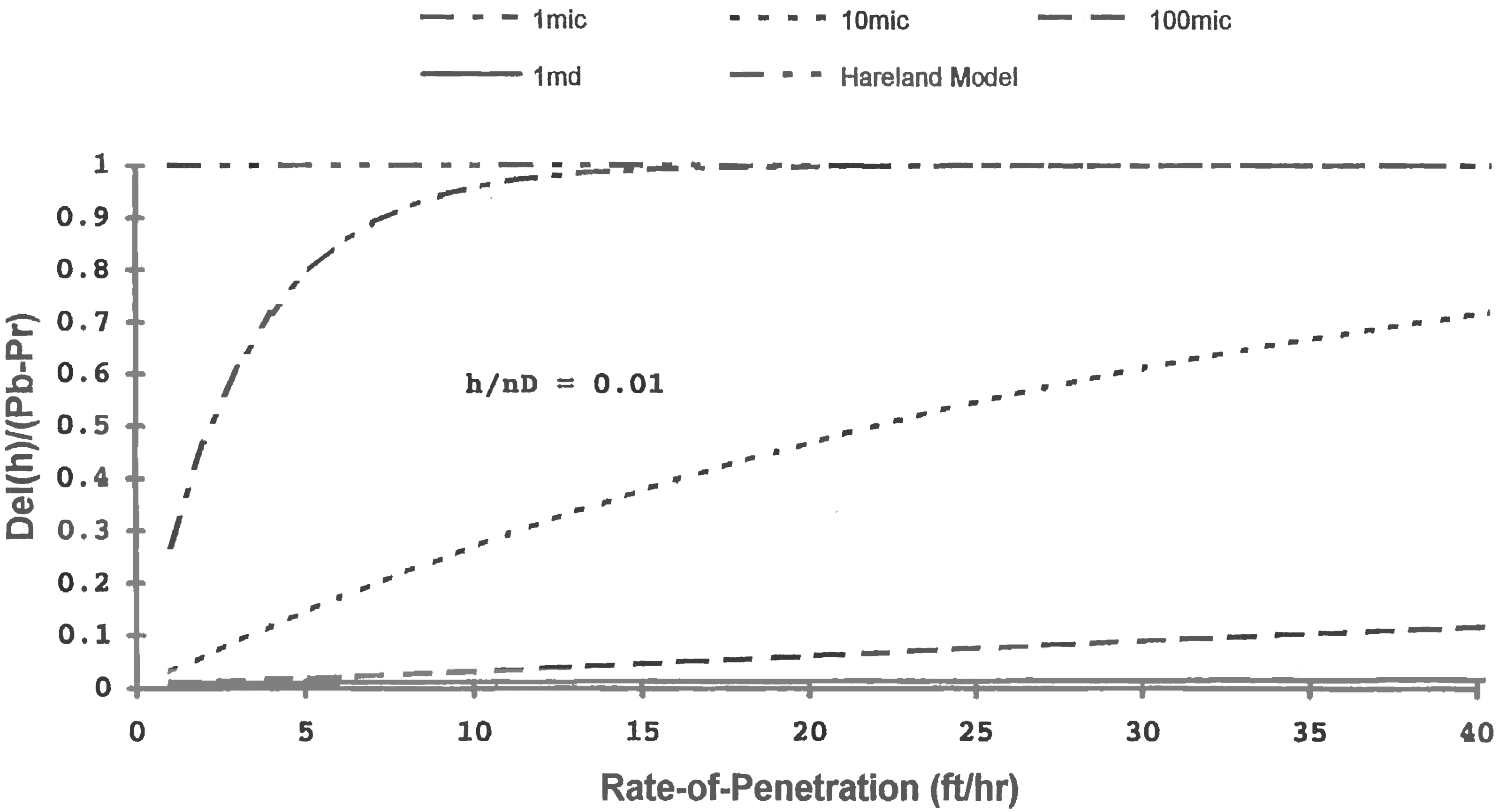


Figure 2.4 Dynamic Differential Pressure Study - nD = 10 in.

where

υ_b = bit velocity (R, rate-of-penetration), ft/hr

$$\upsilon_{fh} = \frac{\alpha}{\psi h} = \frac{k}{\psi \phi c \mu \mathrm{h}} = \text{formation fluid velocity at distance h, ft/hr} \quad (2.28)$$

The ratio of υ_b/υ_{fh} determines how quickly the dynamic differential pressure approaches the maximum differential pressure, P_B - P_R (assumed by Hareland and Hoberock) as a function of R. As υ_b/υ_{fh} increases, the bottom hole advances increasingly faster than the mud filtrate, and $\Delta(h)$ approaches P_B - P_R. Equation (2.28) shows that as k decreases, so does υ_{fh}, allowing less time for the borehole fluid to reach the chip depth h. This results in $\Delta(h)/(P_B\text{-}P_R)$ approaching unity at lower penetration rates as k decreases. In extremely permeable formations, with k large, υ_{fh} will be much larger than υ_b at all penetration rates of interest, and $\Delta(h)$ will approach zero. Available time did not allow testing the effect of dynamic differential pressure calculations on in-situ stress bound calculations. Since confined rock strength decreases as $\Delta(h)$ decreases, this explains why unconfined "strong", but permeable rocks can be drilled faster than unconfined equally "strong", but less permeable rock.

We observe from Figure 2.4 that for rock with permeability of a few microdarcy or less, and for penetration rates of 10 ft/hr or greater, the dynamic differential pressure equals the maximum differential pressure P_B - P_R, as assumed by Hareland and Hoberock. However, for rock with permeability greater than a few microdarcy at virtually all penetration rates, and for penetration rates less than 10 ft/hr at microdarcy or greater permeability, the dynamic differential pressure is significantly less than the maximum assumed by Hareland and Hoberock. Accordingly, for many cases of practical interest in tight gas sands, Hareland and Hoberock's approach uses a confining pressure that is too large, resulting in a predicted rock strength that is too low. This may explain why Hareland and Hoberock's in-situ stress bound prediction for permeable formations generally do not closely match experimental field stress test data. The newly defined formation fluid velocity, υ_{fh}, can be used to determine when Hareland and Hoberock's model is appropriate. Their model can be assumed an appropriate representation of dynamic differential pressure as υ_{fh} becomes small compared to penetration rate.

CHAPTER III

STATIC DIFFERENTIAL PRESSURE FOR IMPERMEABLE AND PERMEABLE FORMATIONS

An Alternative Estimate of Near-Bit Pore Pressure for Impermeable Formations

For impermeable formations, Hareland and Hoberock [1993] assume that effective differential pressure, P_e, equals bottom hold mud pressure, P_B. Effectively, this assumes that as the overburden formation pressure is replaced with pressure due to hydrostatics of the drilling fluid and flowing fluid friction, the formation expands sufficiently to reduce pore pressure, P_p, to zero near the bit. While the pressure on the bottom of the hole due to the drilling fluid weight and flowing friction can be less than half of the overburden pressure, the fluid pressure may nevertheless be sufficient to prevent P_p from reducing to zero. This would invalidate Hareland and Hoberock's assumption, and it seemed appropriate to check this phenomenon with a more detailed analysis than they provided.

Warren and Smith [1985] performed a theoretical study of near-bit pore pressure using a finite element program assuming linear elastic formation behavior, along with equations relating effective stress, σ_e and pore pressure, P_p. They neglected all dynamic effects. According to Nur and Byerlee [1971], effective stress is given by

$$\sigma_e = \sigma - [1 - K_b / K_s] P_p \tag{3.1}$$

where

σ = stress in any direction
K_b = bulk modules of formation
K_s = bulk modules of formation at zero porosity
P_p = pore pressure

Warren and Smith [1985] state that K_b/K_s is typically sufficiently small that (3.1) becomes

$$\sigma_e = \sigma - P_p \tag{3.2}$$

van der Knaap [1959] showed that the bulk volume of a porous rock is a function of the change in mean effective stress, $\Delta\overline{\sigma}_e$, where mean effective stress, $\overline{\sigma}_e$, is calculated as

$$\overline{\sigma}_e = \frac{\sigma_1 + \sigma_2 + \sigma_3}{3} - P_p \tag{3.3}$$

where σ_i, i = 1,2,3, is one of the three principal stresses. As pressure is relieved from the formation, the pore volume increases and P_p decreases, assuming an impermeable formation. Using an equation derived by Geertsma [1957], Warren and Smith [1985] developed a relationship between change in pore pressure, ΔP_p, and $\Delta\overline{\sigma}_e$ as

$$\Delta P_P = \frac{(c_b - c_R)\overline{\Delta\sigma_e}}{\phi C_w + [c_b - (1+\phi)c_R]} \tag{3.4}$$

where

ΔP_P = change in pore pressure
c_b = bulk rock compressibility
c_R = rock matrix compressibility
c_w = pore fluid compressibility
ϕ = porosity
$\Delta\overline{\sigma}_e$ = change in mean effective stress

Given the inter-dependence of $\overline{\sigma}_e$ and P_p, an iterative solution approach was employed by Warren and Smith. To use an iterative approach, suitable initial conditions must be provided. Take σ_1 in the vertical direction and assume it equal to the overburden pressure, using a "nominal" gradient overburden of 1 psi/ft, to obtain

$$\sigma_1 = \sigma_0 = [1.0 \text{ psi/ft}][\text{Borehole depth (ft)}] \tag{3.5}$$

Take σ_2 and σ_3 in horizontal directions and assume a linear elastic isotropic material with equal orthogonal principal stresses, employing a "nominal" horizontal stress gradient of 0.7 psi/ft, to obtain

$$\sigma_2 = \sigma_3 = \sigma_h = [0.7 \text{ psi/ft}][\text{Depth (ft)}] \tag{3.6}$$

An initial value for P_p is taken as a typical salt water gradient times depth, given by

$$P_p = [0.47 \text{ psi/ft}][\text{Depth (ft)}] \tag{3.7}$$

After the overburden is replaced by the mud column, σ_1 becomes

$$\sigma_1 = [M_g][\text{Depth (ft)}] \tag{3.8}$$

where

M_g = mud gradient, psi/ft

With these initial conditions, $\overline{\sigma}_e$ was allowed to change according to (3.3), and P_p changed according to

$$P_p = P_p + \Delta P_p \tag{3.9}$$

where ΔP_p is defined in (3.4).

Given these initial conditions, and assuming M_g=0.47 psi/ft, Warren and Smith determined that P_p at a distance of 0.11 inches into the formation from the center of the borehole would reduce to 1400 psi, or 30 percent of its initial value at a depth of 10,000 ft. This result is contrary to Hareland and Hoberock's assumption of zero pore pressure.

In order to employ Warren and Smith's approach to determine bottom hole pore pressure for impermeable formations, neglecting dynamic effects, a one-dimensional approximation to three-dimensional effects was made. This was necessary because a fully developed three-dimensional finite element program would require a prohibitive amount of computing, given that the computation would be necessary for every foot drilled in a given well. Therefore, a single-element (S.E.) model to approximate near-bit pore pressure at the center of the bore hole was developed. The initial conditions used were those given by Warren and Smith

$$\sigma_{1-} = \sigma_0 = [1.0\text{ psi/ft}][\text{Borehole depth (ft)}]$$

$$\sigma_{1+} = [M_g][\text{Depth}]$$

$$P_{p-} = [0.47\text{ psi/ft}][\text{Depth}]$$

$$\sigma_2 = \sigma_3 = \sigma_h = [0.7\text{ psi/ft}][\text{Depth (ft)}]$$

$$\overline{\sigma}_{e-} = \frac{\sigma_{1-} + \sigma_2 + \sigma_3}{3} - P_p$$

$$\overline{\sigma}_{e+} = \frac{\sigma_{1+} + \sigma_2 + \sigma_3}{3} - P_p$$

$$\overline{\Delta\sigma_{ei}} = \overline{\sigma}_{e+} - \overline{\sigma}_{e-}$$

In the nomenclature above, a subscript negative sign indicates values before overburden removal, and a positive sign indicates values after overburden removal. Only $\overline{\sigma}_e$ and P_p were allowed to change, as determined by (3.3) and (3.9). The resulting algorithm for solving for near-bit pore pressure is as shown in Figure 3.1. The results from the single-element model were then compared with the results reported by Warren and Smith for the center of the bore hole, shown in Table 3.1 and Figures 3.2-3.4 where r/r_w is the ratio of radius to maximum hole radius. Considering the crude grid size of the model, the single-element model shows remarkably good agreement, with typically ten percent or less deviation from the results of Warren and Smith. The computer program EXPAND developed for the single-element model is given in Appendix C.

The single-element program was then inserted as a subroutine in PREDICT (Appendix A), replacing the Hareland and Hoberock's assumption of zero pore pressure for impermeable formations. Data collected during the drilling of SFE#2 was used to calculate an upper bound on in-situ stress. The necessary formation properties were obtained from the topical report on SFE#2 provided by Gas Research Institute (GRI) [1990], and the initial pore pressure gradients for the entire well depth were provided by Ercill Hunt and Associates [1991] and are shown in Table 3.2. The resulting newly calculated upper in-situ stress bounds, labeled "Non-zero P_p Expansion" as well as

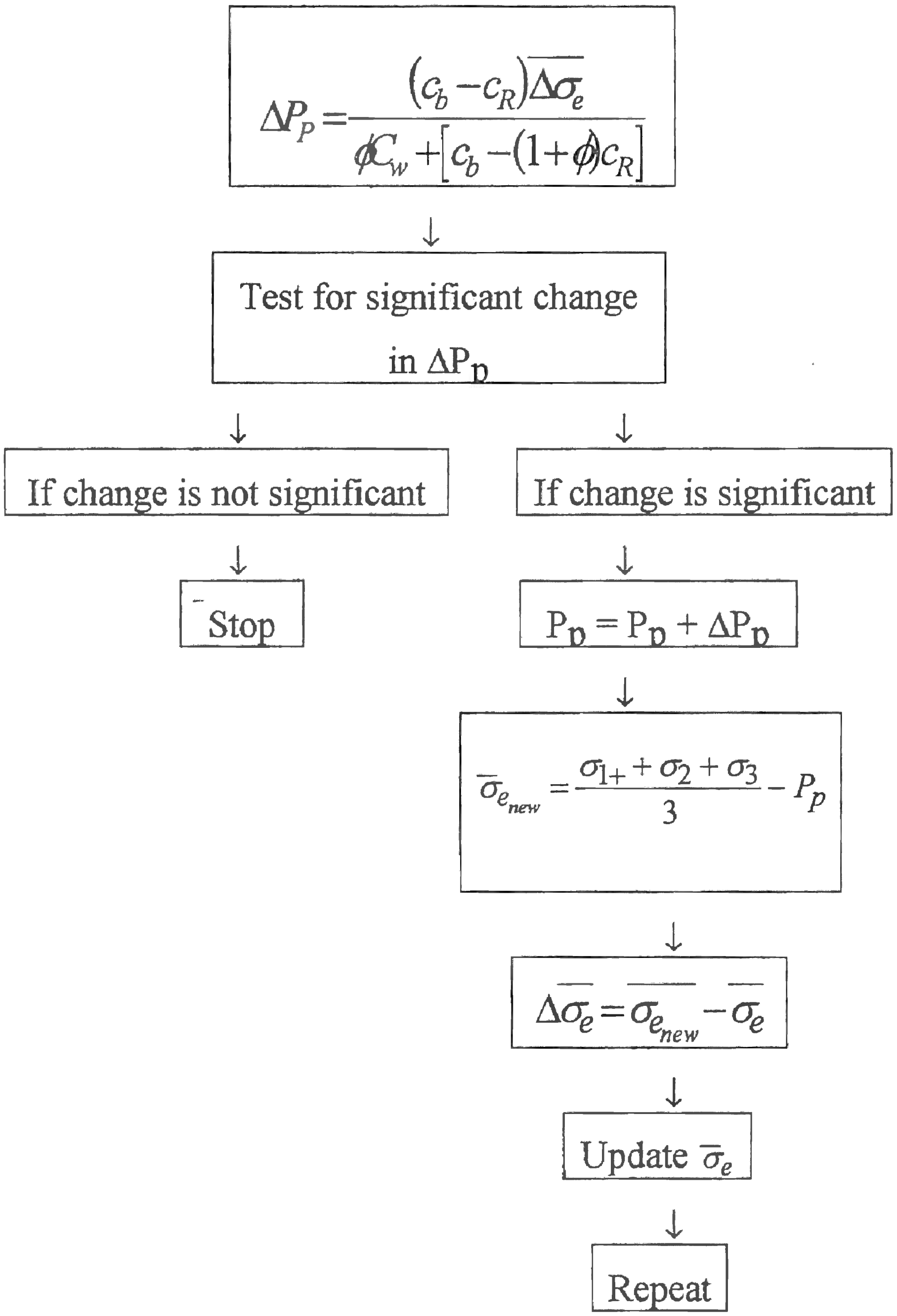


Figure 3.1 - Iteration Procedure for P_p Determination Using the S.E. Model

TABLE 3.1

COMPARISON OF RESULTS OF WARREN AND SMITH [1985] TO THOSE OF SINGLE ELEMENT MODEL

P_{pg} (psi/ft)	M_g (psi/ft)	Depth (ft)	Warren/Smith (psi)	S.E.Model (psi)	%Error
0.47	0.47	5,000	700	680	2.9
0.47	0.47	10,000	1,400	1367	2.4
0.47	0.47	20,000	3,000	2740	8.7
0.47	0.57	10,000	2,000	2034	-1.7
0.47	0.67	10,000	2,800	2700	3.6
0.832	0.78	10,000	0	58	--
0.832	0.832	10,000	450	405	10.0
0.832	0.884	10,000	770	752	2.3

TABLE 3.2

PORE PRESSURE GRADIENT FOR SFE#2

Depth (ft)	Pore Pressure Gradient (psi/ft)
0 to 8,000	0.46
8,000 to 8,800	0.46 + 0.04(Depth - 8,000)/1000
8,800 to 8,900	0.44
8,900 to 9,000	0.46 + 0.04(Depth - 8,000)/1000
9,000 to 10,163	0.50

Provided by Ercill Hunt and Associates

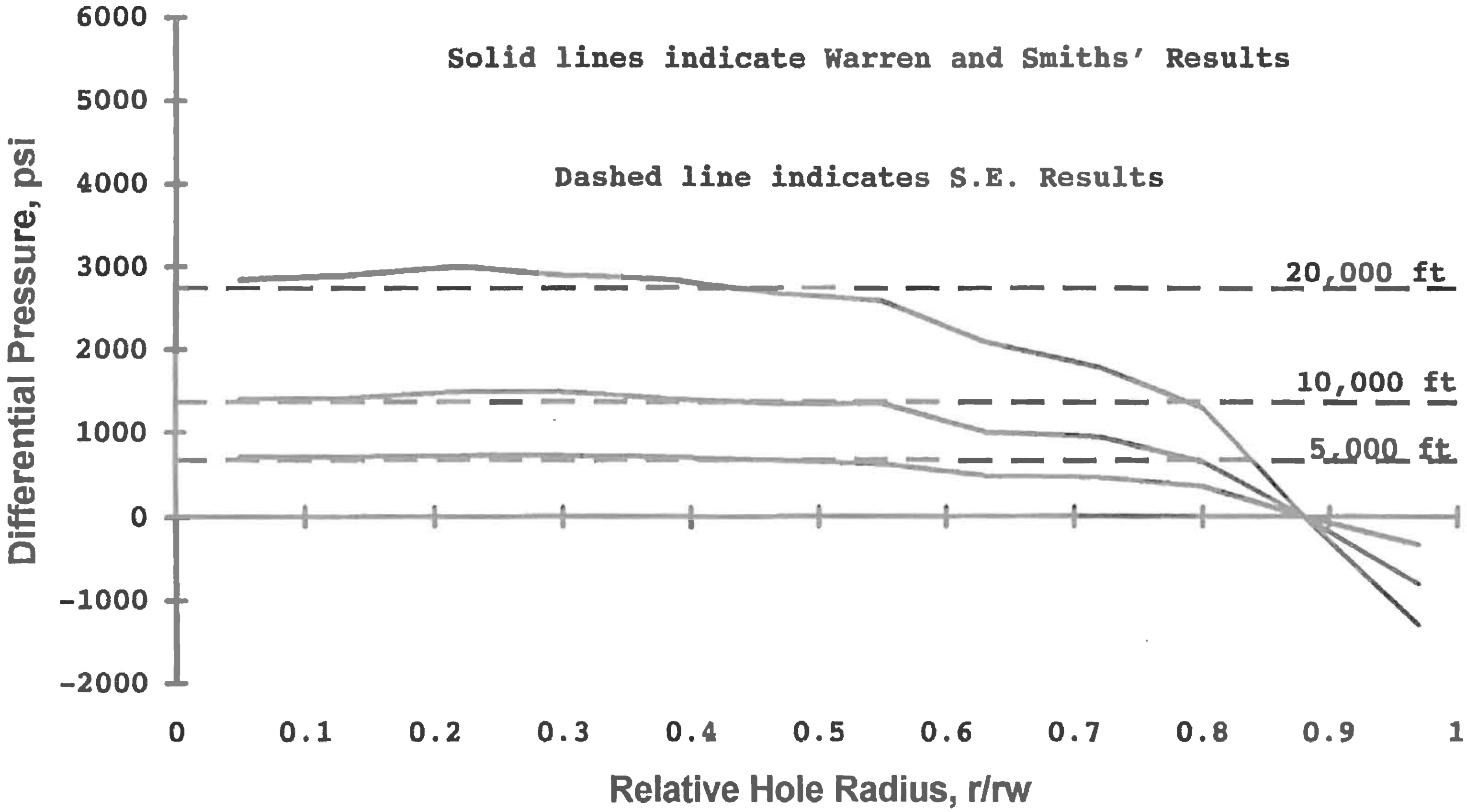


Figure 3.2 Comparison of Results of Warren and Smiths Results to Single Element Model - Various Depths

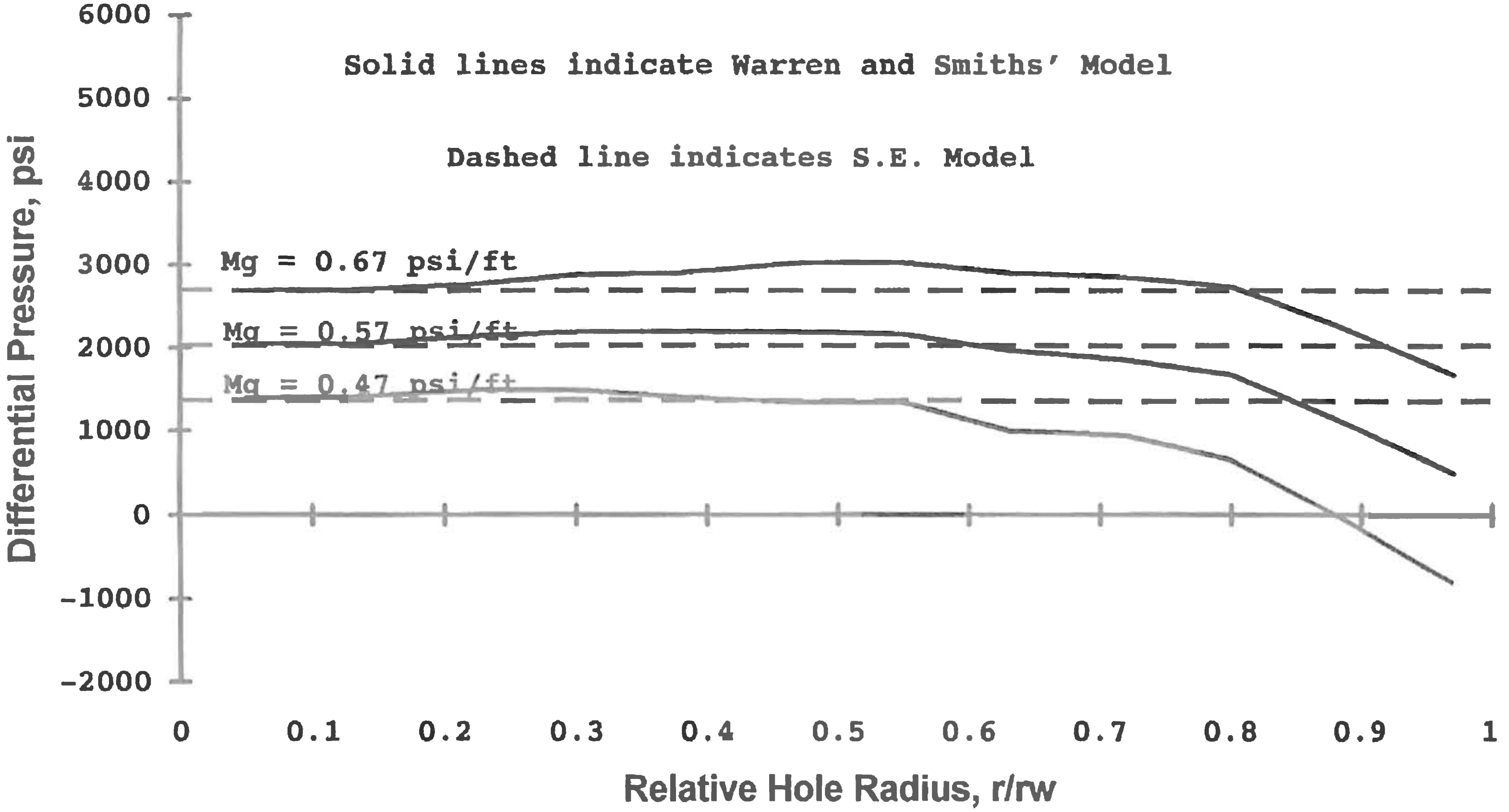


Figure 3.3 Comparison of Results of Warren and Smiths Results to Single Element Model - Various Mg

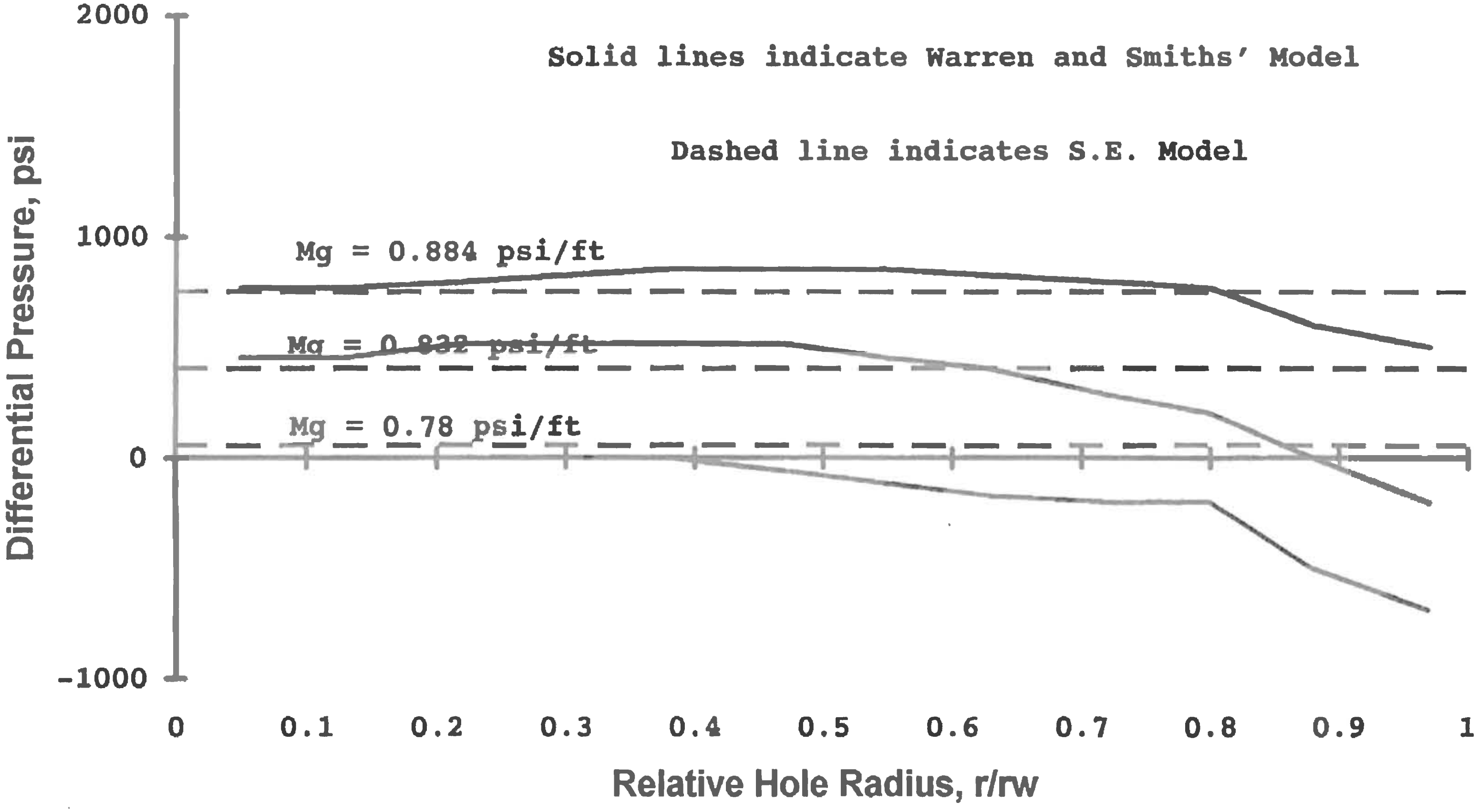


Figure 3.4 Comparison of Results of Warren and Smiths Results to Single Element Model - Various Pp

Hareland and Hoberock's original upper bound, labeled "Zero P_p Expansion", for SFE#2 [1992], are shown in Figures 3.5-3.10.

The figures also show in-situ stress test data as well as several areas containing no upper bound in-situ stress predictions. As the name implies, an in-situ stress test refers to any method for determining the stress as seen by the rock formation. Typically, this test consists of isolating a two foot section of the borehole and slowly increasing the fluid pressure by pumping fluid into the isolated section. The fluid pressure and fluid volumetric flow are continuously recorded during this test and plotted against each other. When the formation fractures, the fluid pressure drops and the volumetric flow of fluid increases. As this continues, ideally the fluid pressure verses volumetric flow plot levels out at a single pressure. This pressure is taken as the formation closure pressure. Formation closure pressure is the pressure at which the fracture can be maintained open, but further fracture propagation does not occur and, typically, is assumed equal to the in-situ stress. The areas in Figures 3.5 to 3.10 containing no upper bound predictions are due to absence of appropriate drilling data needed to generate a prediction using PREDICT. PREDICT assumes the use of a roller cone bit, while the areas in question were drilled using a coring bit.

The upper bound with non-zero P_p expansion does, at several depths where stress tests were performed, provide a closer approximation to in-situ stress than the upper bound with zero P_p expansion, as can be seen at depths of 8960, 9760, 10,025 and 10,110 ft. However, as an overall upper bound on in-situ stress, the upper bound calculated with the S. E. model fails. Note that Hareland and Hoberock's upper bound prediction lies above or equal to virtually all of the in-situ stress test data points, while the upper bound calculated using the S. E. model does not.

There appears to be a reasonable explanation for why this new approach does not show improvement over Hareland and Hoberock's upper bound calculation, namely, that the "differential pressure" effect on drilled chip removal remains poorly understood. Their

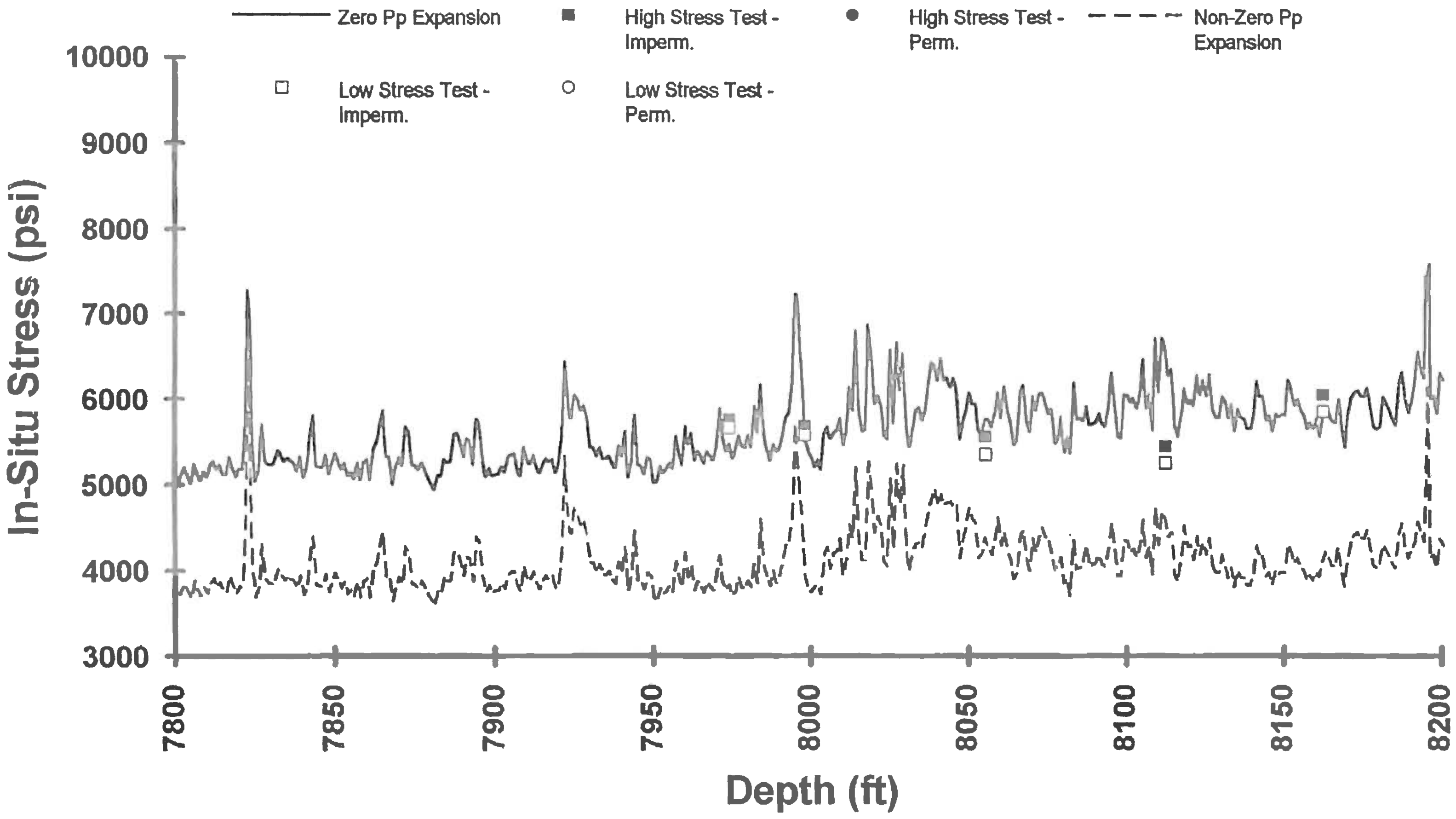


Figure 3.5 Comparison of Impermeable Stress Bounds - Zero Vs. Non-Zero Pp Bottom Expansion
7800-8200 Ft. Depth

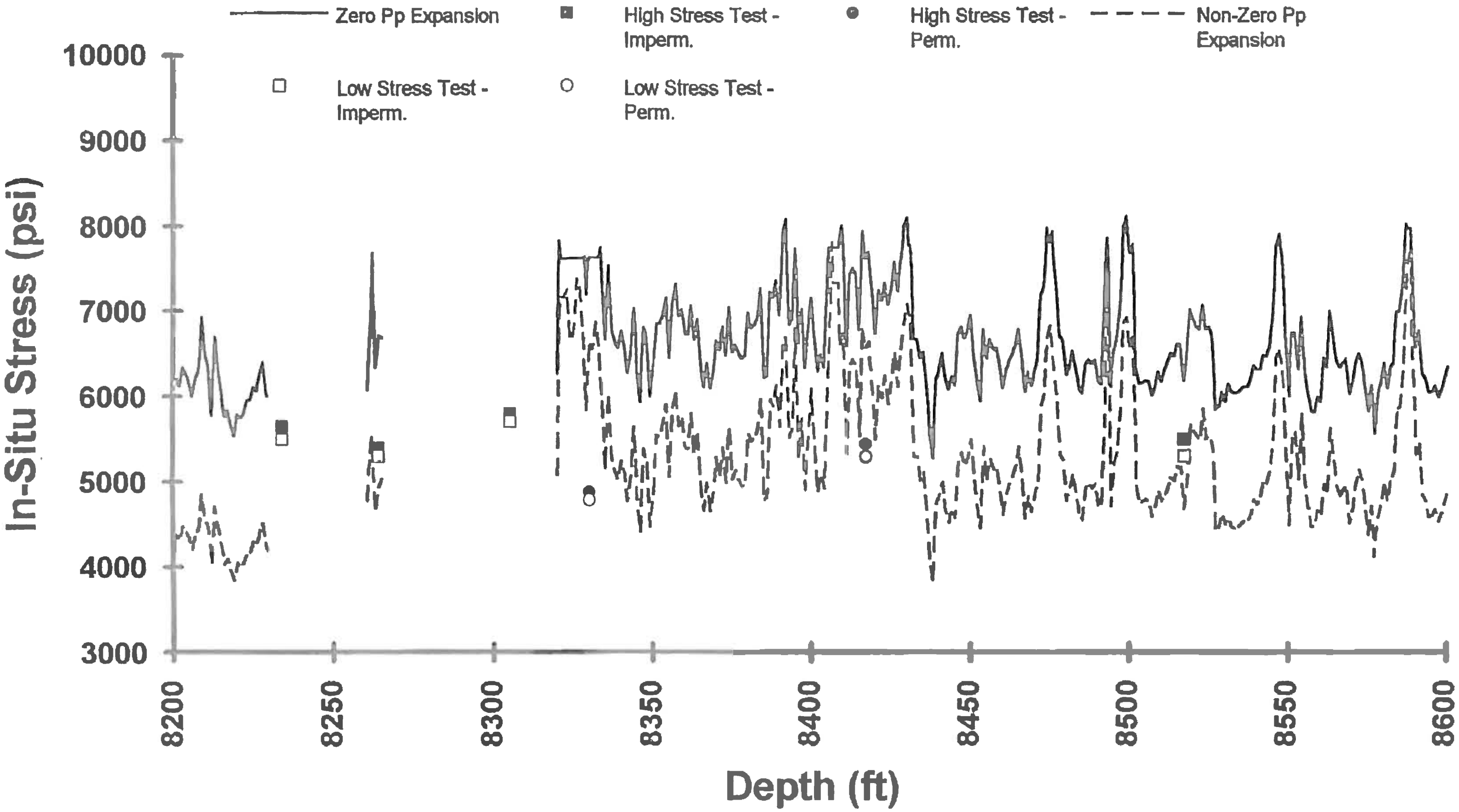


Figure 3.6 Comparison of Impermeable Stress Bounds - Zero Vs. Non-Zero Pp Bottom Expansion 8200-8600 Ft. Depth

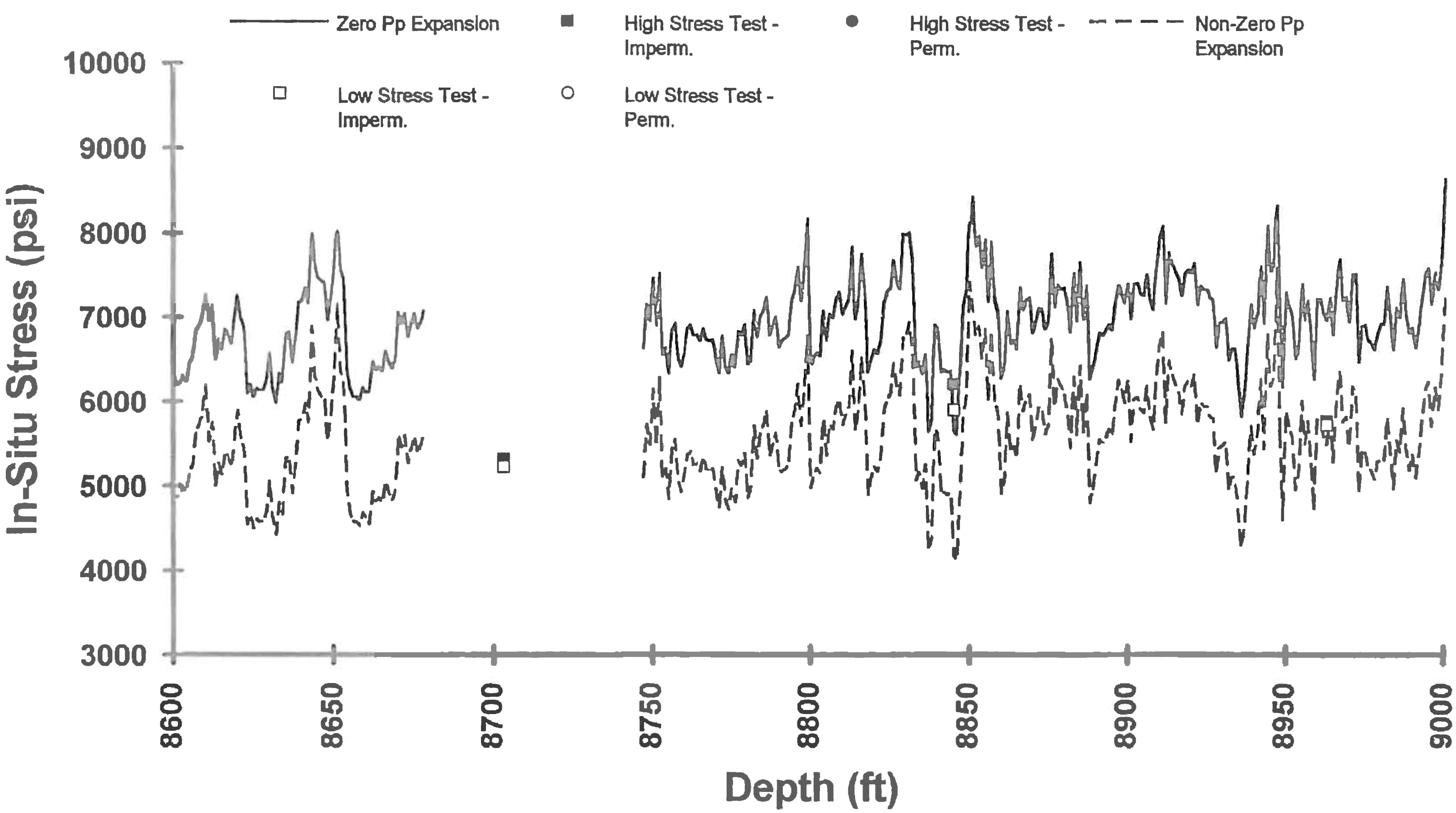


Figure 3.7 Comparison of Impermeable Stress Bounds - Zero Vs. Non-Zero Pp Bottom Expansion
8600-9000 Ft. Depth

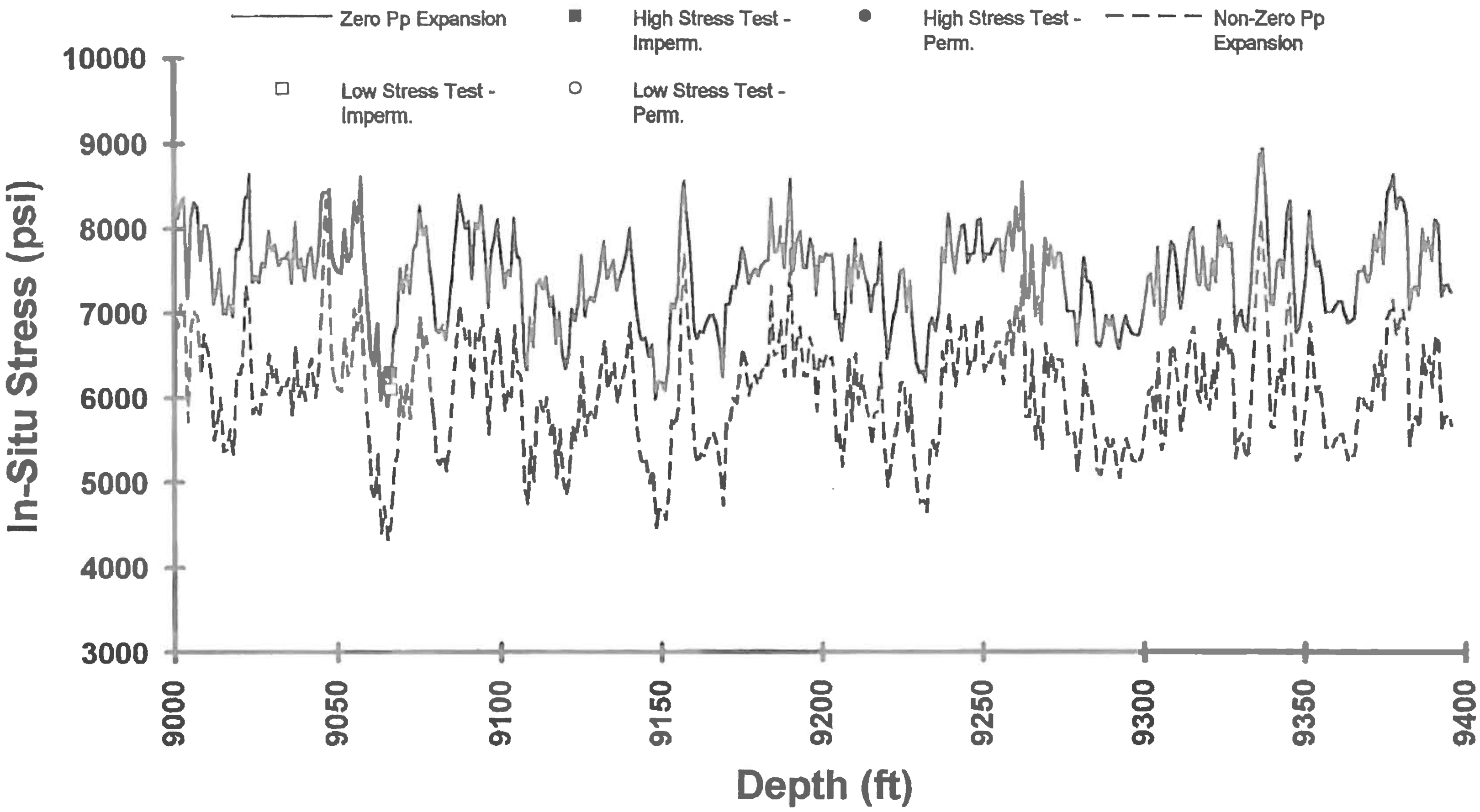


Figure 3.8 Comparison of Impermeable Stress Bounds - Zero Vs. Non-Zero Pp Bottom Expansion 9000-9400 Ft. Depth

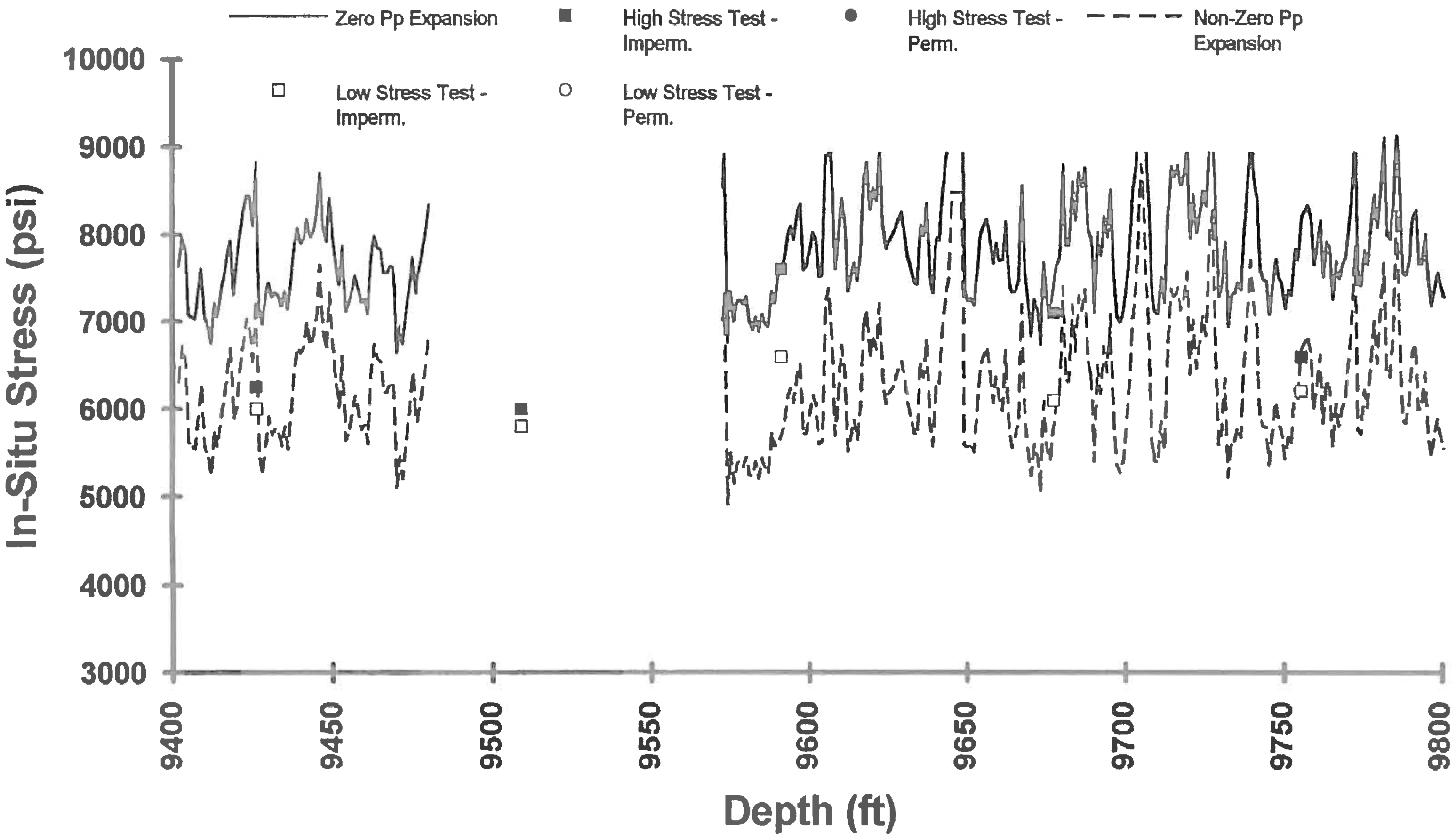


Figure 3.9 Comparison of Impermeable Stress Bounds - Zero Vs. Non-Zero Pp Bottom Expansion 9400-9800 Ft. Depth

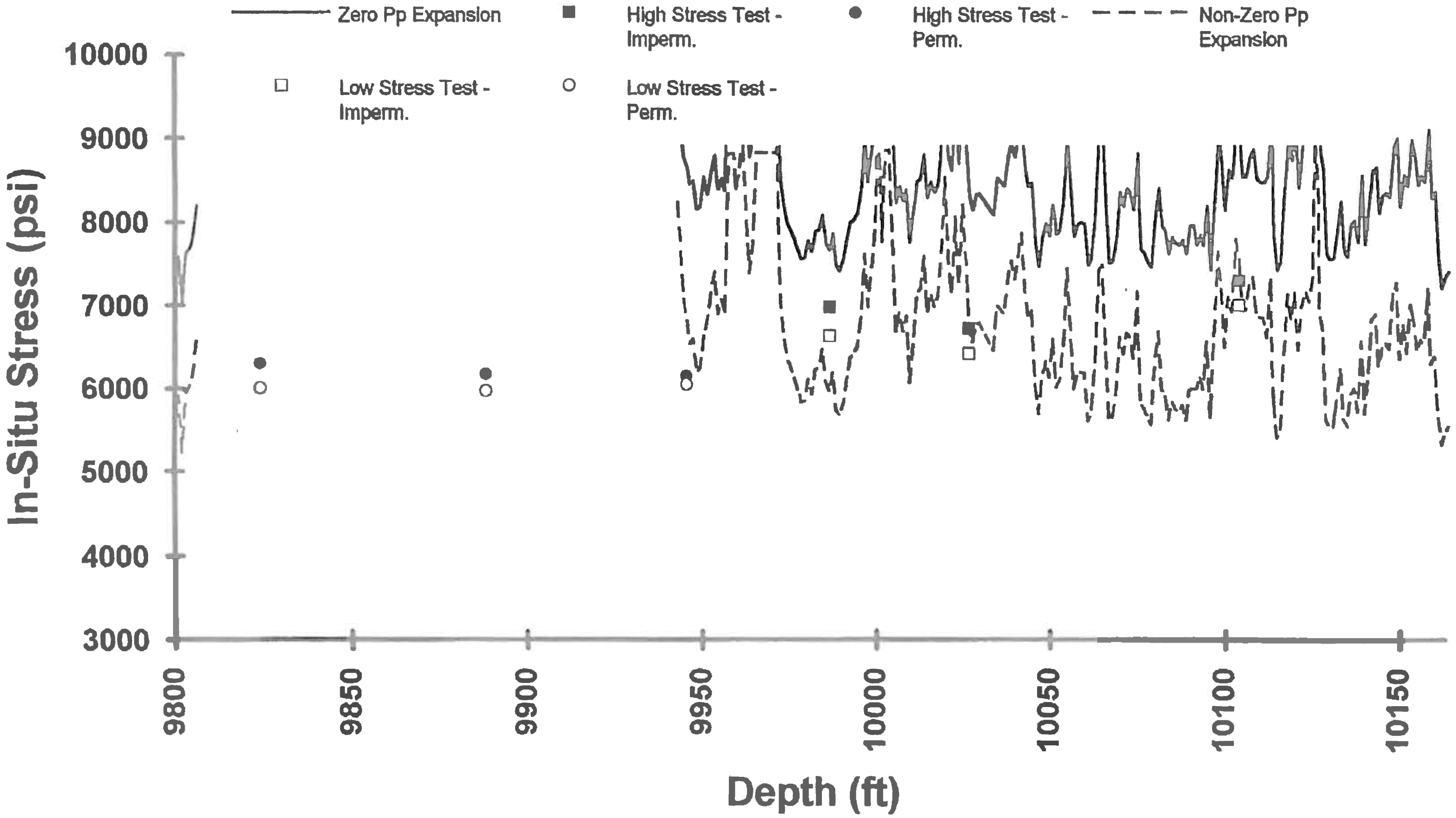


Figure 3.10 Comparison of Impermeable Stress Bounds - Zero Vs. Non-Zero Pp Bottom Expansion 9800-10160 Ft. Depth

method, while giving good results, nevertheless is an empirical approach, using a data-fitted "chip hold-down function". On the other hand, while Warren and Smith present an appealing physical theory and computational model for near-bit pore pressure changes, they ignore both dynamic effects and the effect of chip formation and crack propagation into the formation as the bit advances. It is possible that while Warren and Smith's approach may be correct and appealing for undamaged formation near the bit, chip formation and crack propagation change near-bit pore pressure and drilling fluid pressure effects in such a way that only drilling fluid pressure acts to effectively "confine" the drilled rock, as Hareland and Hoberock empirically assume. Until more careful and detailed laboratory studies are performed to understand this phenomenon, it seems unlikely that better predictive methods for in-situ rock strength and formation stress will be developed using drilling penetration rate models.

Minimum Differential Pressure for Permeable Formations

Equation (1.2) assumes differential pressure is affected by pore pressure. This relationship has been traditionally used to define confining pressure while drilling sandstone, which is a more permeable formation, drilling faster than shale under identical drilling conditions, even though sandstone is normally a stronger formation. Borehole and pore fluid are assumed to have minimal communication due to the formation of mud cake, which is the build-up of drilling mud solids on the bore hole surface. For mud cake to form, the drilling mud experiences a certain amount of spurt-loss, which is the initial mud filtrate loss to the formation during cake build-up. Considering the depth of interest is approximately one bit tooth penetration, spurt loss may be significant enough for pressure equalization across a drilled chip. This would yield a low value of differential pressure.

To test this theory, PREDICT (Appendix A) was used to calculate the upper bound on in-situ stress in a permeable formation with the differential pressure set equal to 120 psi, the lowest allowable differential pressure under Hareland and Hoberock's method. The results are given in Figures 3.11-3.16, with the label "zero P_e" used for results from this calculation and the label "non-zero P_e" used for results from Hareland and Hoberock's calculation, all for permeable formations. Note that the two upper bounds are very close for all depths. The new boundary is lowered by only a few psi over most of the well depth.

A possible explanation for this insensitivity is the inadequate modeling of differential and pore pressure in Hareland and Hoberock's method. Differential pressure is limited to a minimum of 120 psi in PREDICT. It may be that in permeable formations, actual differential pressure is in fact never more than a few hundred psi, and that Hareland and Hoberock's assumption is close. No means are presently available to set differential pressure to zero in their model and test this effect.

As the figures indicate, no general improvement in the permeable or impermeable upper bound has been obtained by either method in this chapter for determining P_e. Therefore, while Hareland and Hoberock's approach for determining P_e is not validated, their approach appears to be a reasonable approximation to a poorly understood phenomenon.

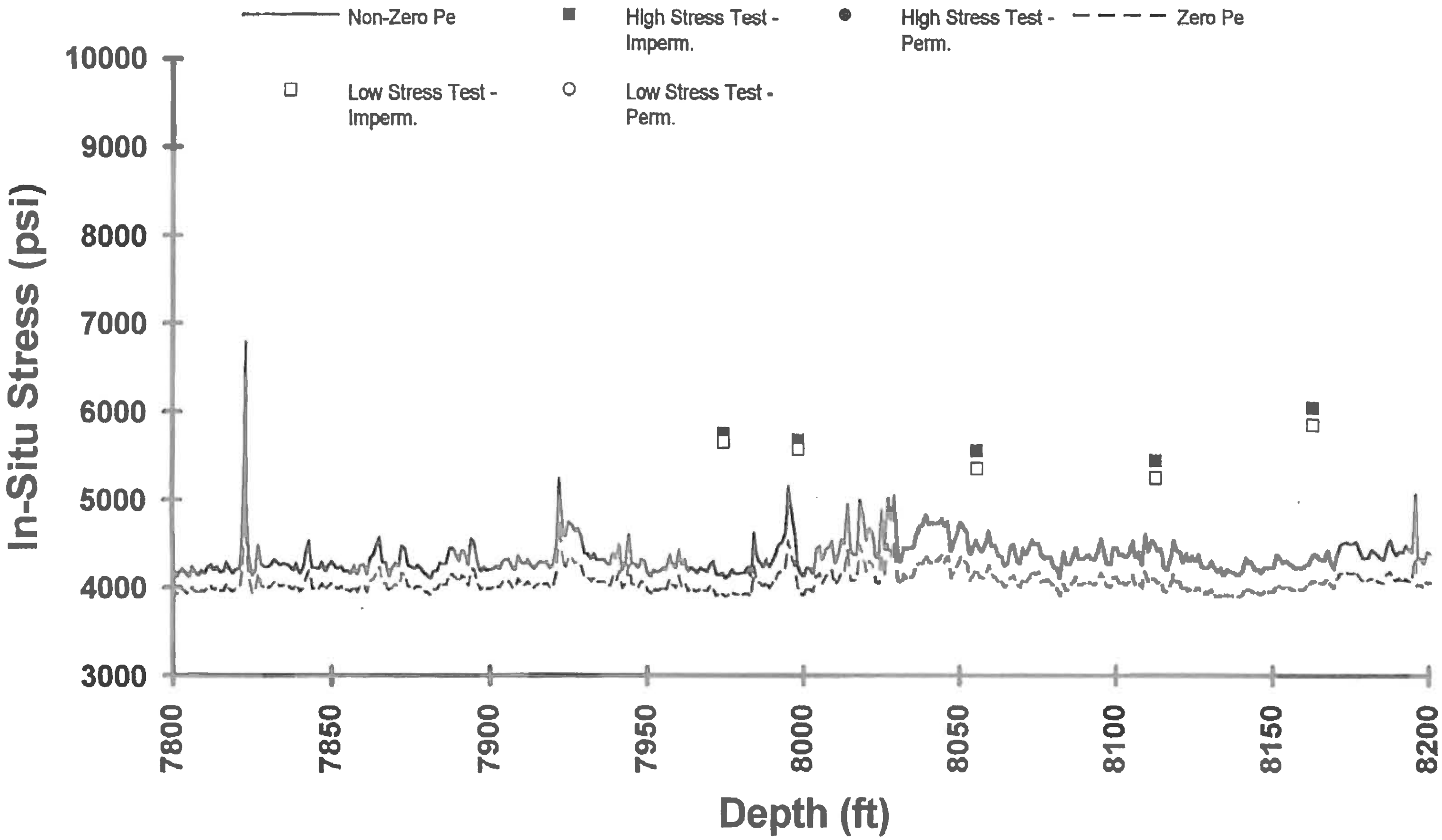


Figure 3.11 Comparison of Permeable Stress Bounds - Zero Vs. Non-Zero Confining Pressure
7800-8200 Ft. Depth

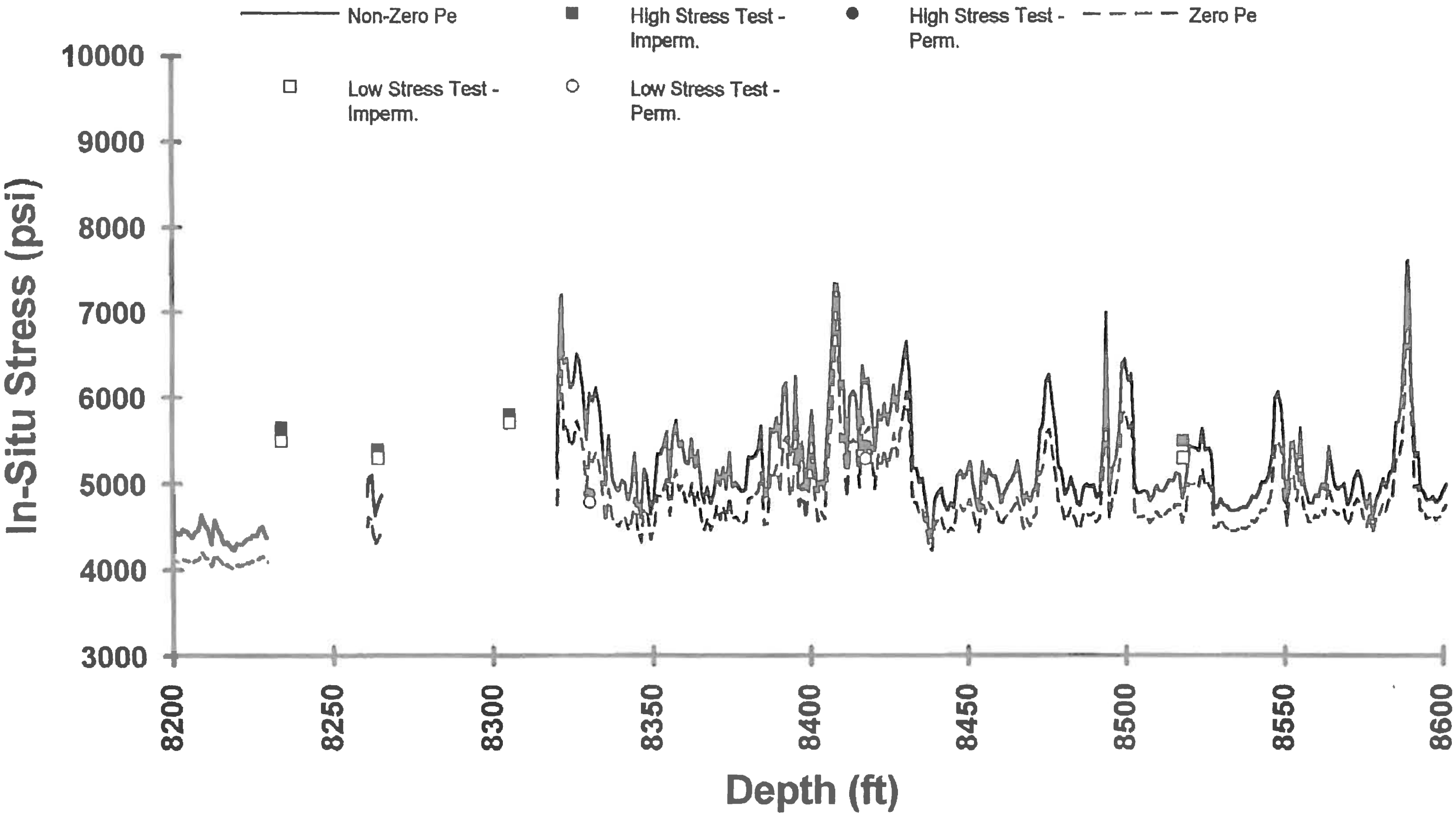


Figure 3.12 Comparison of Permeable Stress Bounds - Zero Vs. Non-Zero Confining Pressure
8200-8600 Ft. Depth

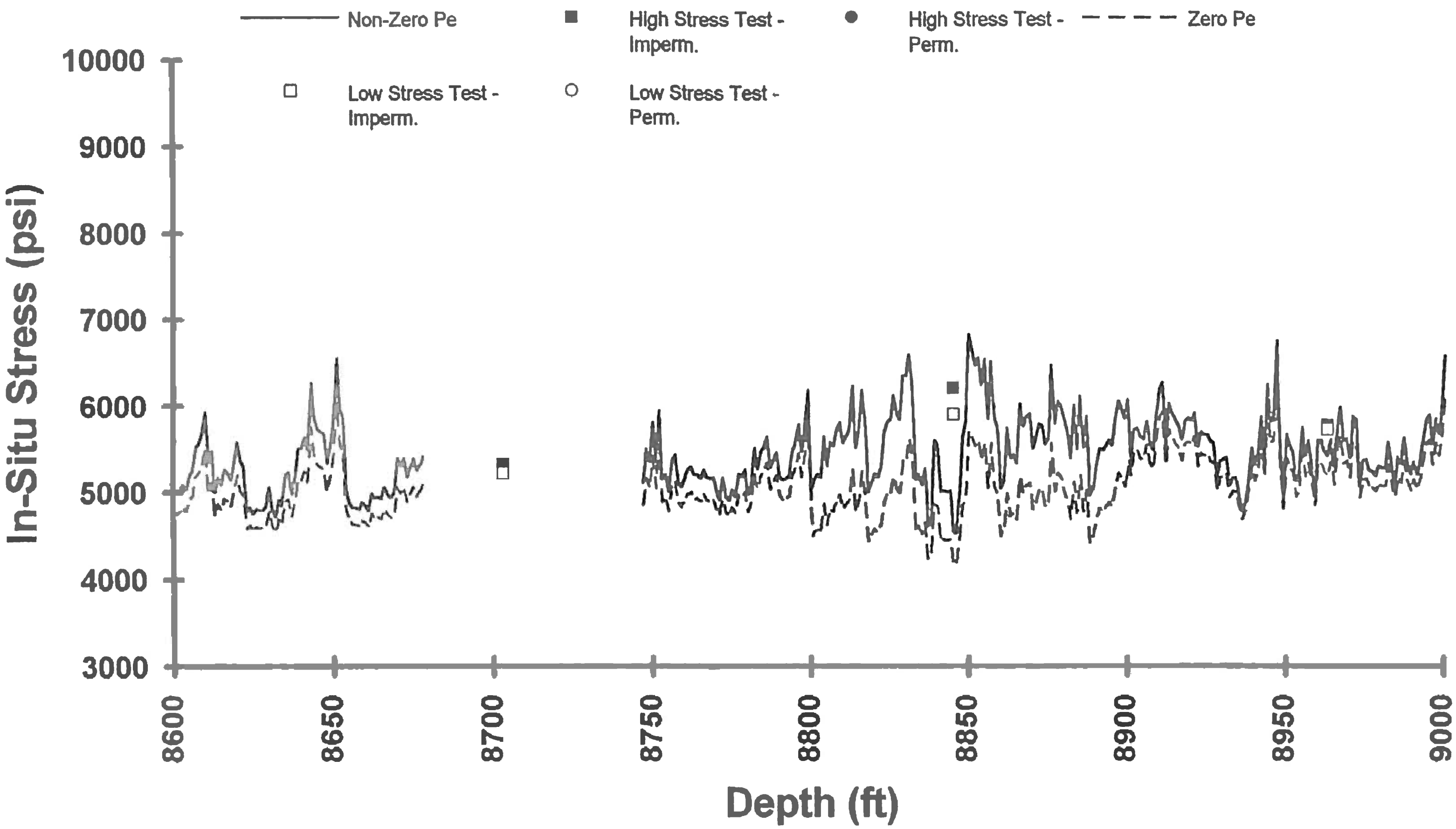


Figure 3.13 Comparison of Permeable Stress Bounds - Zero Vs. Non-Zero Confining Pressure 8600-9000 Ft. Depth

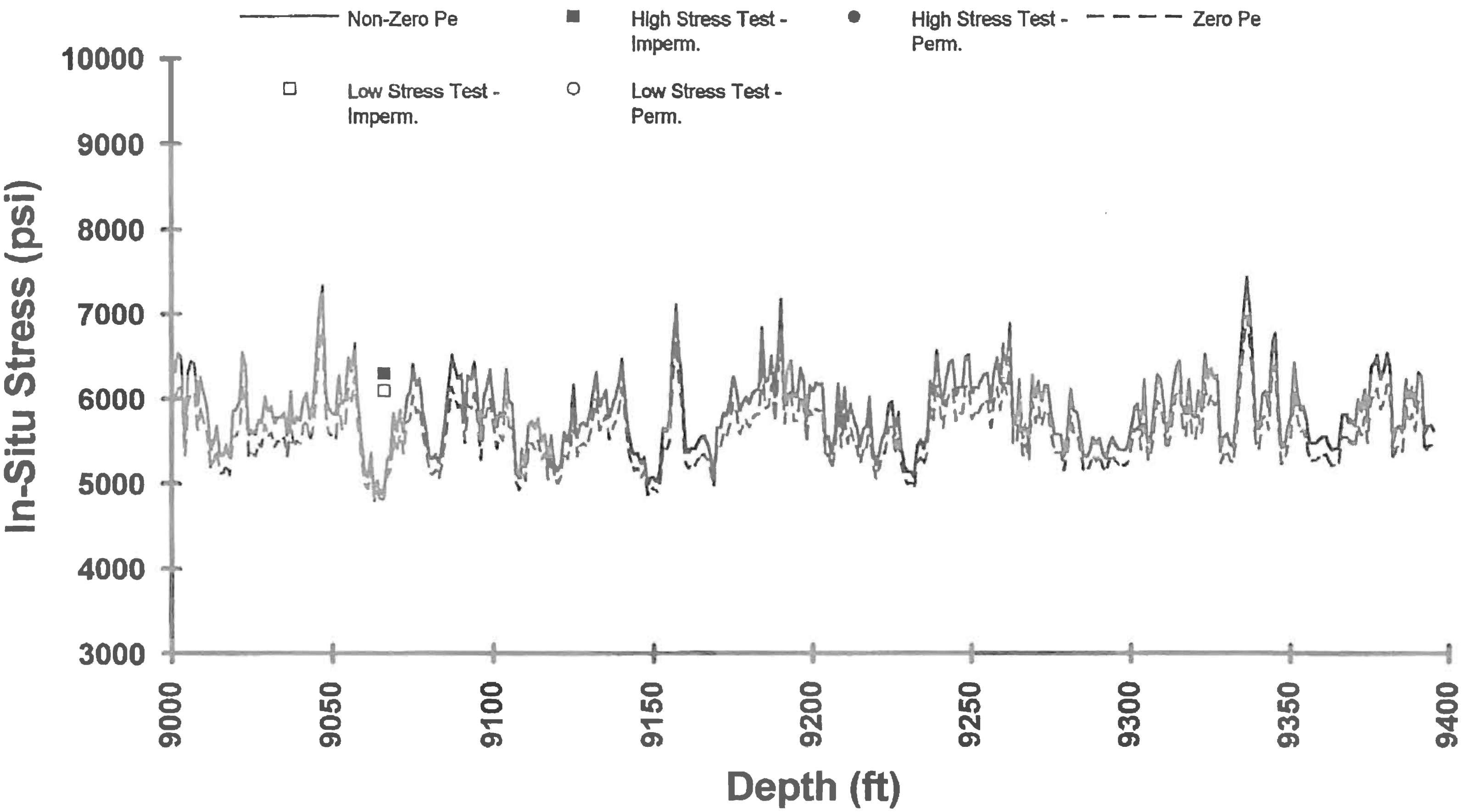


Figure 3.14 Comparison of Permeable Stress Bounds - Zero Vs. Non-Zero Confining Pressures
9000-9400 Ft. Depth

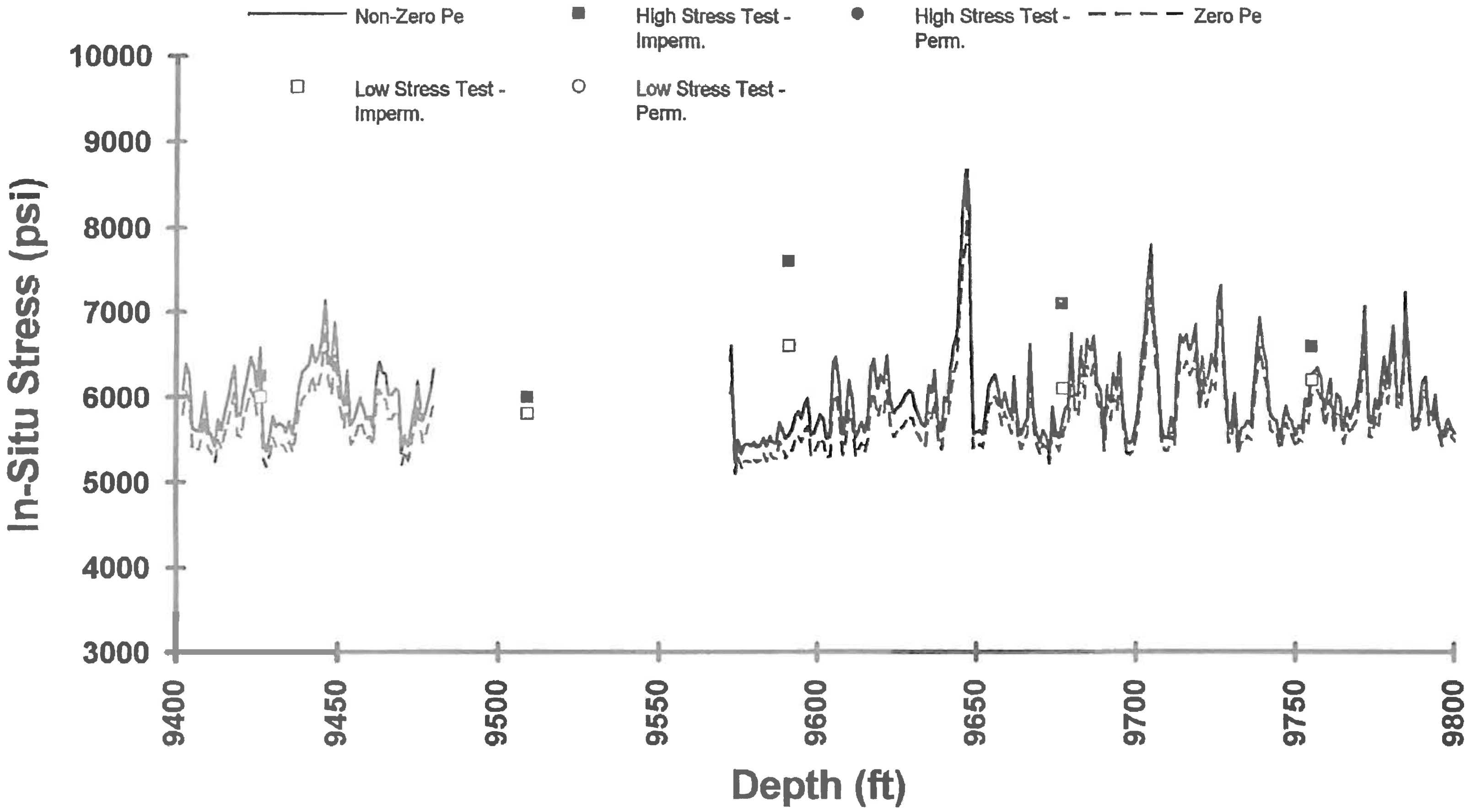


Figure 3.15 Comparison of Permeable Stress Bounds - Zero Vs. Non-Zero Confining Pressure
9400-9800 Ft. Depth

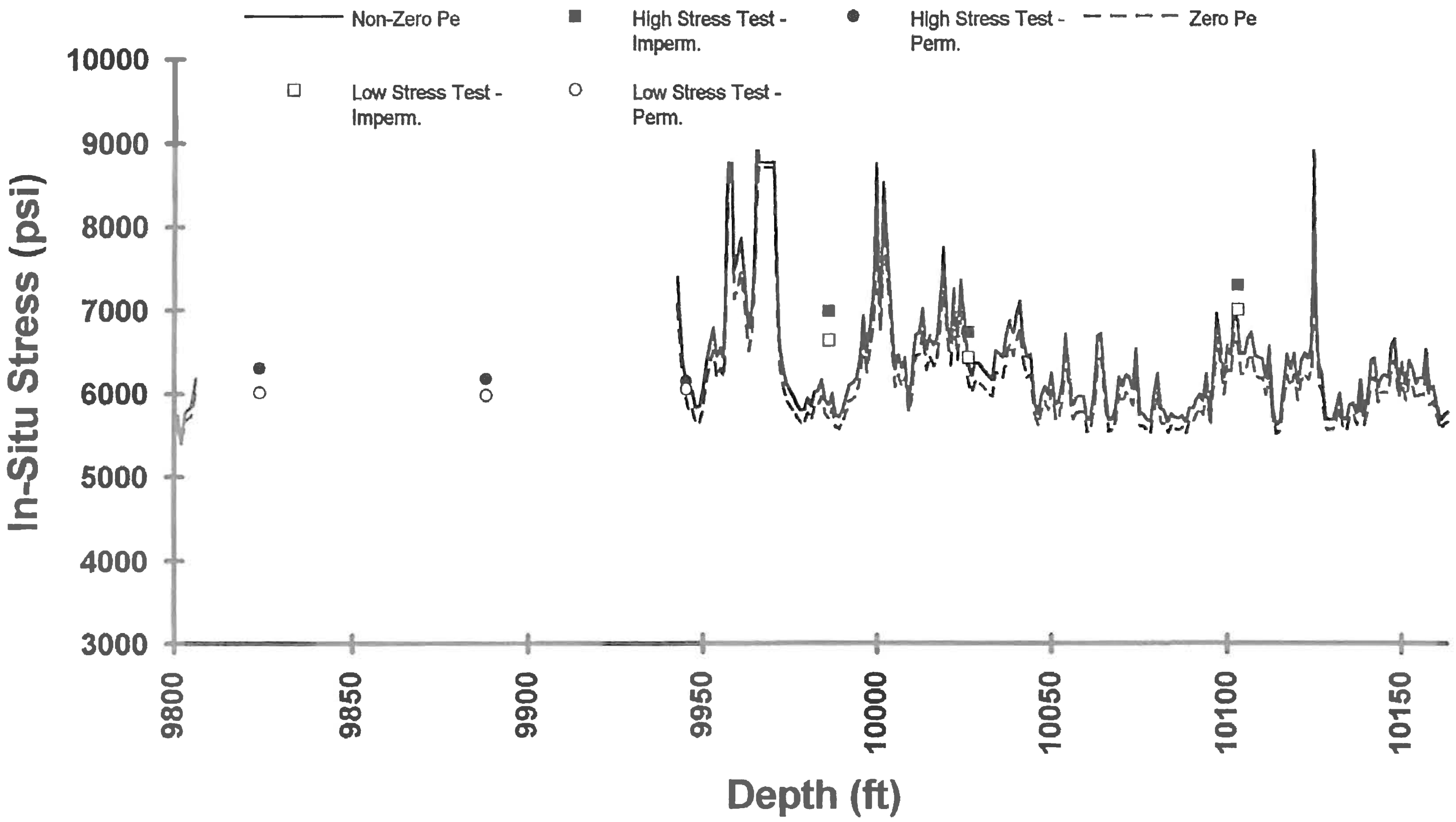


Figure 3.16 Comparison of Permeable Stress Bounds - Zero Vs. Non-Zero Confining Pressure
9800-10160 Ft. Depth

CHAPTER IV

A NEW APPROACH FOR DETERMINING ROCK STRENGTH

The equation used by Hareland and Hoberock [1993] to determine in-situ rock strength, σ, is given as

$$\sigma=\left[\frac{NW^2}{af_c(P_e)RD^3}-\frac{bW^2}{aD^4}-\frac{c\rho\mu NW^2}{af_c(P_e)I_mD^2}\right]^{1/2} \tag{4.1}$$

where

σ = Compressive rock strength, psi
N = Rotary speed, rpm
W = Weight-on-bit, lbf
$f_c(P_e)$ = Chip hold-down function, unitless
R = Rate-of-penetration, ft/hr
D = Bit diameter, inches
ρ = Drilling mud density, lbf/gal.
μ = Drilling mud viscosity, cp
I_m = Modified impact force, lbf
a, b = Bit design coefficients, (hr rpm in)/ft
c = Bit design coefficients, (hr lbf gal)/(ft lbm cp in)

As discussed in previous chapters, this equation contains a number of empirical terms, including the expressions for effective confining pressure, P_e, the chip hold-down function, $f_c(P_e)$, and the three bit design coefficients, a, b, and c. Moreover, penetration rates measured on the drilling rig are subject to high degrees of fluctuation due to measurement techniques, drill string dynamics, and the drilling process itself. If a high-fidelity model based on the physics of the process were available, and accurate drilling data could be easily measured and recorded, much of the uncertainty could be eliminated.

Teale [1965] proposed an equation based on an energy balance of a drilling tool. Defining specific energy, E_s, as work divided by volume drilled, Teale derived the following expression for E_s

$$E_S = \frac{F}{A} + \left(\frac{2\pi}{A}\right)\left(\frac{NT}{u}\right) \tag{4.2}$$

where

E_S = Specific energy, psi
F = Force applied to the drilling tool, lbf
A = Surface area of the drilling tool as seen by the drilled surface, in^2
N = Rotary speed of the drilling tool, rpm
T = Rotary torque applied to the drilling tool, in-lbf
u = Penetration rate of drilling tool into drilled surface, in/min

The first term in (4.2) represents the thrust or crushing specific energy, while the second term represents the specific energy due to rotation. Pessier and Fear [1992] applied (4.2) specifically to a well drilling bit, obtaining

$$E_S = \frac{W}{A_B} + \frac{120\pi NT}{A_B R} \tag{4.3}$$

where

W = Weight-on-bit, lbf
A_B = Area of the bottom of the hole, in^2
R = Rate of penetration, ft/hr
T = Bit torque, ft-lbf

Both Teale and Pessier and Fear noted that when E_S is at its minimum, E_{sm}, it approximates the compressive strength of the formation being drilled, that is

$$E_{sm} = \sigma, \text{ psi} \tag{4.4}$$

This suggests that, given accurate drilling data, (4.3) could provide an approximation of in-situ compressive rock strength without the need to determine bit coefficients or a chip hold-down function. This would constitute a substantial simplification over the approach used by Hareland and Hoberock.

Equation (4.3) may not always be suitable for field use. While calibrated values for W, N, R, and A_B are usually measured, only relative values for T are typically recorded. Of course, calibration of the torque sensor can be easily accomplished, as suggested in Appendix D, but this is not a common practice. Pessier and Fear [1992] provide an equation to calculate torque from commonly recorded drilling data with the expression

$$T = \frac{\tau D W}{36} \tag{4.5}$$

where τ is a unitless coefficient of sliding friction and, for most roller cone bits, ranges from 0.2 to 0.25 with 0.21 as an average. Upon substitution, (4.3) becomes

$$E_s = \frac{4W}{\pi D^2} + \frac{13.33\tau N W}{DR} \tag{4.6}$$

where we have used $A_B = \pi D^2/4$, with D being the bit diameter, in^2.

Employing τ does introduce some uncertainty into (4.3). However, Pessier and Fear [1992] have determined that, for a roller-cone bit in good condition, with good hydraulics for cleaning, τ is fairly constant. For more precise calculations, a procedure for calibrating a torque sensor on the rig has been developed and successfully tested by Hoberock, Hareland, and the author. Appendix D contains an outline of this procedure.

Now consider the problem of how the minimum value E_{sm} of E_s in (4.3) or (4.6) can be determined to find σ in (4.4). Apparently, for a given rock strength, we should vary relevant drilling parameters over an appropriate range of operating conditions, calculating E_s for each set of parameters, and select the minimum value of E_s as E_{sm}. For this, we propose a widely used field test, called a "drill-off test" (DOT). In this simple experiment, the drilling brake is locked while the rotary table continues to turn the bit at a constant speed, N. As the bit advances by stretching the drill pipe, weight on bit gradually decreases, or is "drilled-off", and penetration rate also decreases. Usually, a drill-off test consumes only a few feet of formation, such that formation rock strength is less likely to vary significantly during the test. Of course, for this procedure to be useful, such rock strength must remain essentially constant during the test.

Because the drilling brake is locked during a drill-off test, such that the drilling hook, or top of the drill string, is stationary, penetration rate, which is typically measured at the hook, will measure zero. Fortunately, however, penetration rate at the bit can be calculated during a DOT using a "pipe stretch model", given by Bourgoyne et al. [1986] as

$$R = -\left(\frac{3600L}{EA}\right)\frac{dW}{dt} \tag{4.7}$$

where

L = Length of the drill string, approximately borehole depth, ft
E = Average modulus of elasticity of the drill string, psi
A = Average cross-sectional area of the drill string, in^2
t = Time, sec

This relationship assumes that all of the weight indicated by the weight on bit indicator on the fig floor is applied at the bit. Accordingly, it applies only to straight, vertical holes with no "tight spots", "ledges", or other interference between bore hole and drill string. As mentioned earlier, in normal drilling, field measured values of R fluctuate widely when recorded second-by-second, or even minute-by-minute. Hence, a potential side benefit of using (4.7) is the elimination of the fluctuation in R. Unfortunately, measured values for dW/dt are subject to similar wide fluctuations. However, values measured for W and t are considerably less erratic. Therefore, if a suitable expression relating W and t could be found, (4.7) could be used directly in (4.3) or (4.6).

Given the positive results Hareland and Hoberock [1993] showed using (4.1) with field data, (4.1) can be reasonably assumed to model penetration rate and, therefore, can be assumed to represent the relationship between W and R. Substituting the right side of (4.7) for R in (4.1) yields

$$-\left(\frac{EA}{3600L}\right)\frac{dt}{dW} = f_e\left(P_e\right)\left[\frac{a\sigma^2 D^3}{NW^2} + \frac{b}{ND}\right] + \frac{c\rho\mu D}{I_m} \tag{4.8}$$

Assuming that formation properties are constant during a DOT, integration of (4.8) yields

$$(t-t_0)=\frac{K_1}{N}\left(\frac{1}{W}-\frac{1}{W_0}\right)+\frac{K_2}{N}(W_0-W)+K_3(W_0-W) \qquad (4.9a)$$

where

t_0 = starting time of drill-off test, sec
W_0 = weight on bit at start of drill-off test, lbf

$$K_1=\frac{3600La\sigma^2D^3f(P_e)}{EA}, \quad \text{assumed constant} \qquad (4.9b)$$

$$K_2=\frac{3600Lbf(P_e)}{EAD}, \quad \text{assumed constant} \qquad (4.9c)$$

$$K_3=\frac{3600Lc\mu\rho D}{I_mEA}, \quad \text{assumed constant} \qquad (4.9d)$$

Equations (4.9) are valid only for drilling in a drill-off test, under the assumptions noted above. Assuming a drill-off test is conducted over a minimal depth change, K_1, K_2, and K_3 are positive constants, which provides a continuous relationship between t and W. By comparing (4.1) and (4.9), relationships can be inferred between each K_i and a specific drilling action. K_1 is a function of single tooth penetration, since the first term in (4.1) models this [Warren, 1987]; K_2 relates to multiple teeth penetration effects; and K_3 relates to the hydraulic cleaning term.

The relative importance of each term in (4.9) is shown in Tables 4.1 and 4.2 for some typical drilling data. The values for K_i in Table 4.1 were calculated using actual drilling data collected from Staged Field Experiment #4 at different depths according to (4.9b), (4.9c), and (4.9d). The values in Table 4.2 were calculated using the values for the K_i given in Table 4.1, assuming values for W and W_0 of 40,000 lbf and 30,000 lbf, respectively, and a value for N of 60 rpm. As illustrated in the tables, the first term in (4.9a) containing K_1 has the greatest magnitude, with the K_2 term an order of magnitude smaller, and the K_3 term almost insignificant. This conforms with intuition, since K_3 is related to the hydraulic cleaning term, which is almost always the least significant term in (4.1). Accordingly, in the procedures below, we will set K_3 equal to zero.

TABLE 4.1

ORDER OF MAGNITUDE STUDY FOR K_i IN EQUATION (4.9)

Depth (ft)	K_1 (rpm lbf sec)	K_2 (rpm sec / lbf)	K_3 (sec / lbf)
7284	1072101000.0	0.093077	0.000169
7683	3256790000.0	0.093894	0.000212
7745	5192691000.0	0.094945	0.000205
7830	2431079000.0	0.097080	0.000186
7956	6995575000.0	0.097874	0.000229
8007	3436021000.0	0.102502	0.000225

According to (4.9), the K_i can be explicitly calculated if the necessary drilling data is available. However, in what follows, we will not attempt to calculate values for the K_i as given by (4.9b - 4.9d), but rather, will determine them directly from DOT data.

Taking derivatives with respect to W in (4.9a) yields

$$\frac{dt}{dW} = -\left[\frac{K_1 + K_2 W^2 + K_3 W^2 N}{N W^2}\right] \tag{4.10}$$

Substitution in (4.6), using the results for R in (4.7) and (4.10), gives

$$E_s = \frac{4W}{\pi D^2} + \frac{13.33\tau EA}{3600LD}\left[\frac{K_3 W^2 N + K_2 W^2 + K_1}{W}\right] \tag{4.11}$$

After rearrangement, (4.11) becomes

TABLE 4.2

ORDER OF MAGNITUDE STUDY FOR TERMS IN EQUATION (4.9)

	N = 60 rpm	W=30,000 lbf	W_0=40,000 lbf
Depth	$K_1/N^*(1/W-1/W_0)$	$K_2/N^*(W_0-W)$	$K_3(W_0-W)$
(ft)	(sec)	(sec)	(sec)
7284	148.903	15.513	1.69
7683	452.330	15.649	2.12
7745	721.207	15.824	2.05
7830	337.650	16.180	1.86
7956	971.608	16.312	2.29
8007	477.225	17.084	2.25

$$E_s = \left[\frac{13.33\,\tau EAK_1}{3600LD}\right]\left(\frac{1}{W}\right) + \left[\frac{4}{\pi D^2} + \frac{13.33\,\tau EAK_2}{3600LD}\right](W)$$

$$+\left[\frac{13.33\,\tau EAK_3}{3600LD}\right](WN) \qquad (4.12)$$

It can now be seen that for any value of W, E_s must achieve its minimum when N = 0. Setting N = 0 in (4.12), define E_{so} as

$$E_{so} = \left[\frac{13.33\,\tau EAK_1}{3600LD}\right]\left(\frac{1}{W}\right) + \left[\frac{4}{\pi D^2} + \frac{13.33\,\tau EAK_2}{3600LD}\right](W) \qquad (4.13)$$

For given values of K_i, (4.13) can be used to plot E_s verses W, as shown in Figure 4.1, in parameters in Table 4.3. This figure illustrates how E_{so} varies with W and that the minimum value E_{sm} of E_{so} (and E_s) occurs at a critical weight on bit, W_c. W_c is determined by setting $dE_{so}/dW = 0$ in (4.13), and solving for W_c. The result is

$$W_c = \left[\frac{13.33\,\tau D \pi E A K_1}{14400L + 13.33\,\pi D \tau E A K_2} \right]^{1/2} \tag{4.14}$$

Given values for K_1, K_2, and K_3, determined as shown below from DOT data, W_c from (4.14) can be used for W in (4.13) to find E_{sm}, which approximates the in-situ rock strength.

In order to evaluate (4.13) and (4.14), drill-off test data from bit performance tests (BPT) on SFE#4 was used in a nonlinear regression routine, SAS [1992], to determine the values for the K_i in (4.9a). A bit performance test consists of several DOT's conducted in rapid succession. The collected data was extensively edited to eliminate obviously erroneous data. Such erroneous data consisted of extremely low or high weights-on-bit, or data which illustrated physically impossible trends, such as increasing weight on bit with time. As an example, Figure 4.2 shows unedited DOT data from BPT #8, with W plotted versus t. Note, for example, the suspicious data points near times 45200, 45400, and 46000 sec. Obviously, if rock formation properties are constant over a drill-off test, W must continuously and smoothly decrease with time during a drill-off test, according to (4.9a). After editing, the plot of data in Figure 4.2 reduces to that given in Figure 4.3.

There was insufficient good data available over a depth range of constant formation properties to allow values of all three K_i to be determined. This does not pose a significant problem, since, in magnitude, the term containing K_3 is the least significant contributor in (4.9a) as illustrated in Tables 4.1 and 4.2. By setting $K_3 = 0$, values for K_1 and K_2 could be determined by the program SAS from edited DOT data. These values were then used to calculate E_{sm} and W_c for each BPT, and the values for $\sigma = E_{sm}$ were compared with values provided by Hareland [1992].

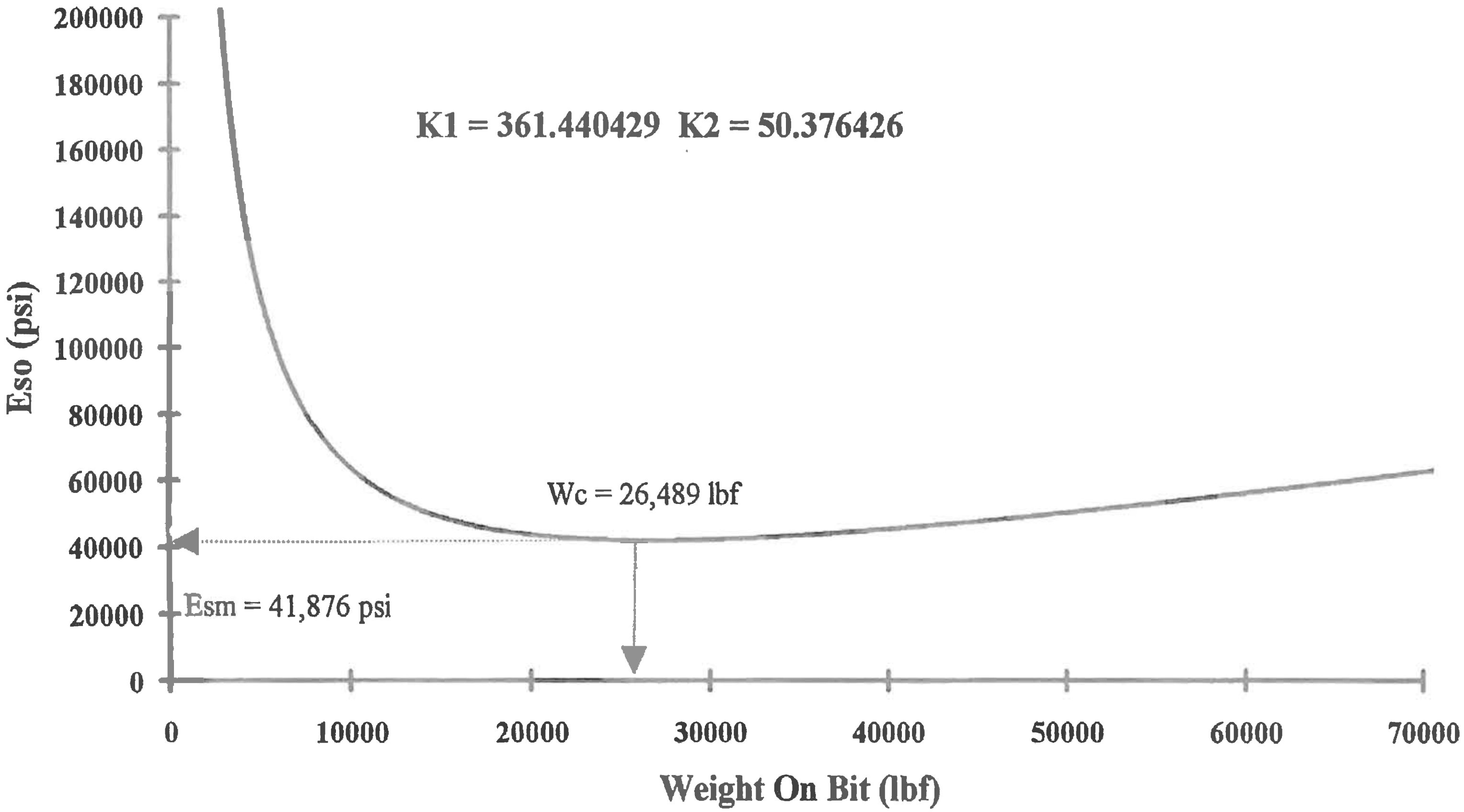


Figure 4.1 Specific Energy using K_i for BPT#8 at Zero Rotary Speed

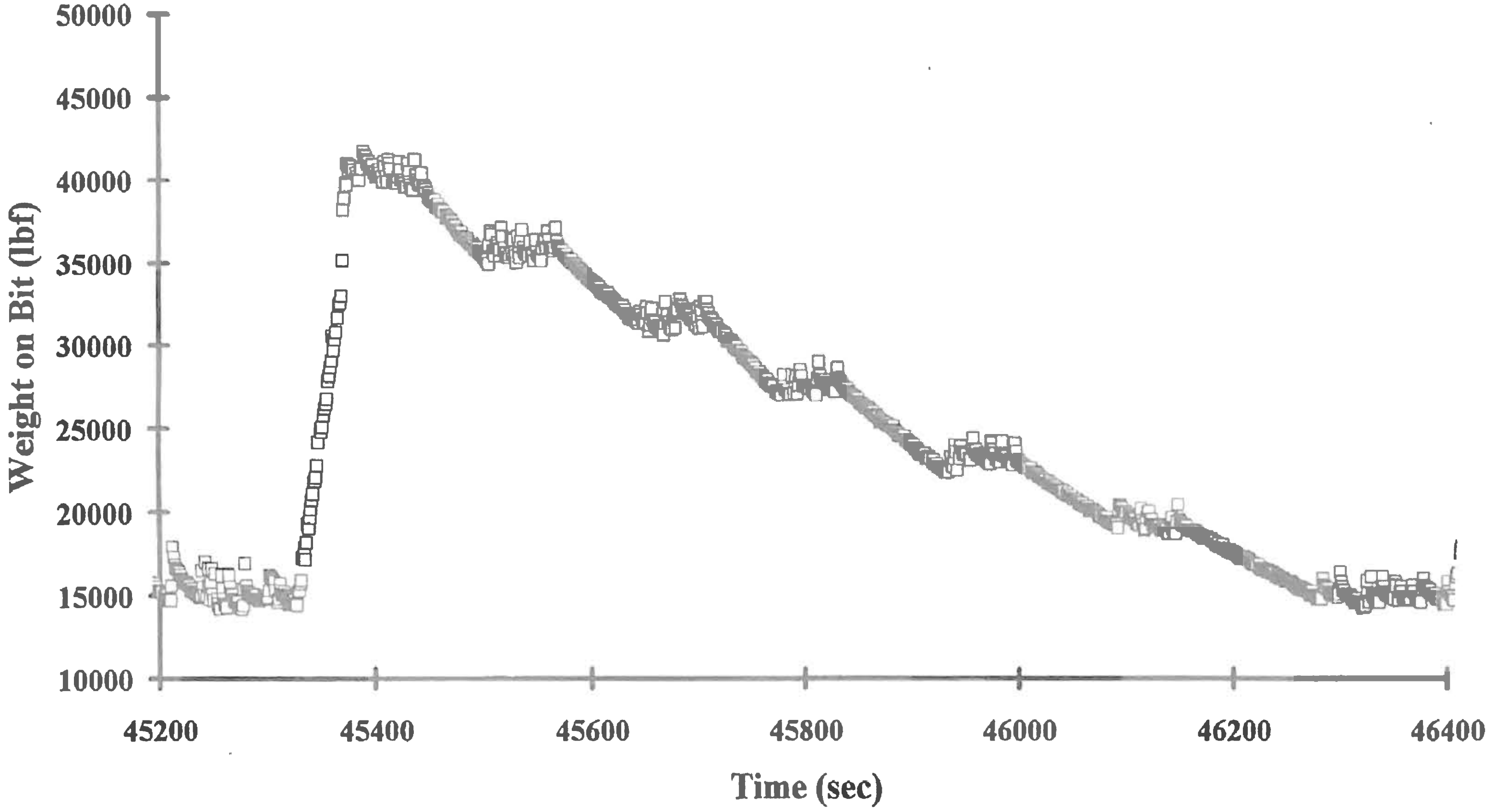


Figure 4.2 Unedited Data Collected for BPT#8

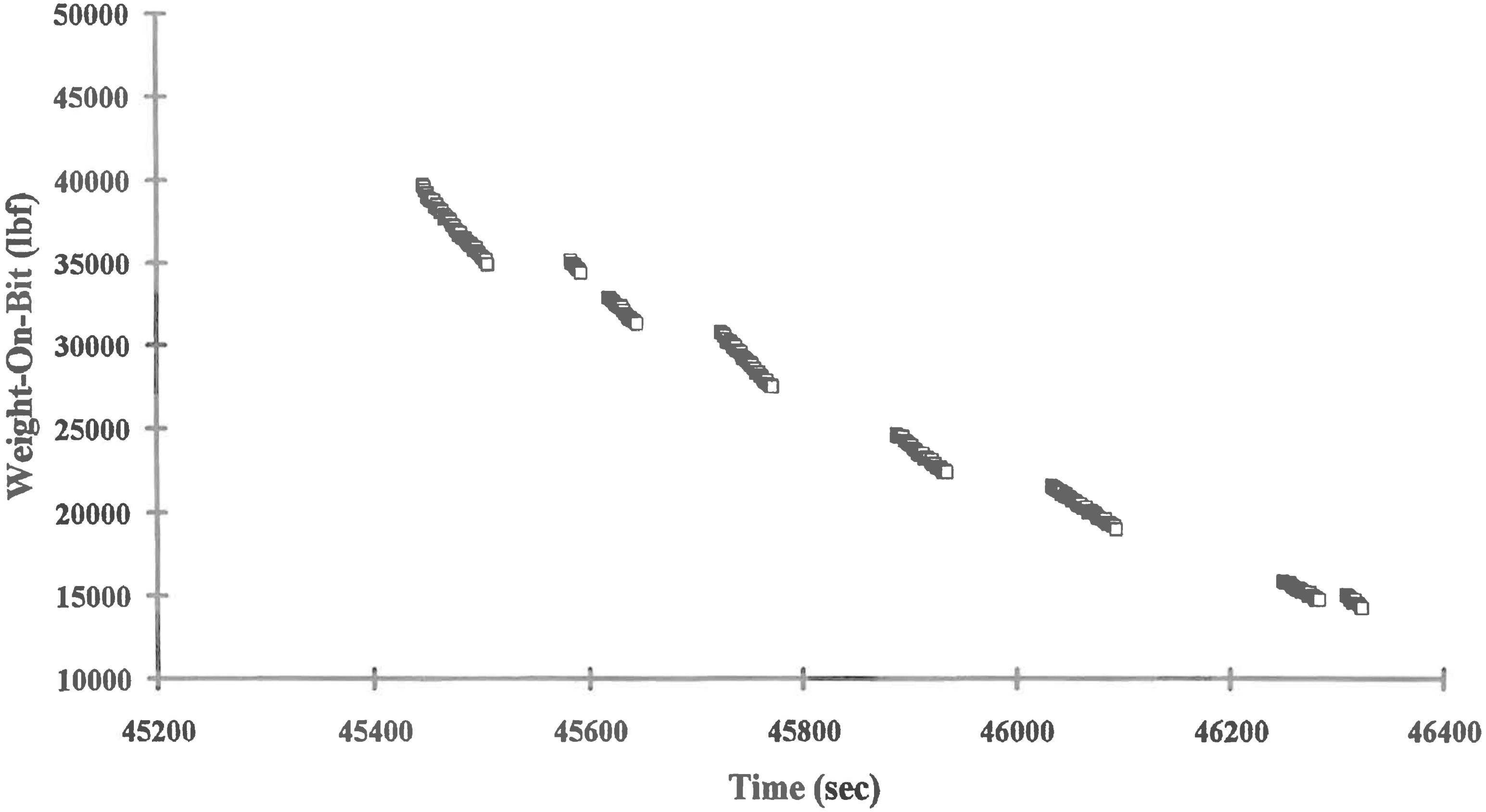


Figure 4.3 Edited Data Collected for BPT#8

Initially the K_i were determined using the DOT data as a continuous set, i.e., t_0 and W_0 were set at the start time of the DOT and these values were used for the entire test. Results are shown in Figure 4.4, with values of τ, D, E, A, and L given in Table 4.3. While the curve using the SAS-calculated values for K_i and edited DOT data does approximate the edited the DOT data, significant anomalies are apparent. The approximating curve tends to track the centers of the individual data segments in Figure 4.4, as is expected in a least-squares curve fit. However, in most cases the approximating curve under-estimates the beginning of each data segment and over estimates the data at the end of each data segment. Engineering judgment suggests that the method used to calculate values for K_i in Figure 4.4 has not produced a reasonable fit to the data.

The method developed in this chapter assumes continuous drill-off test data. While each segment of data in Figure 4.4 appears to represent a short continuous DOT, when taken as a whole, the data are not continuous. This may be due to intermittent "hanging" of the drill string on borehole ledges or tight spots, such that the full weight is not applied to the bit in these tight spots. Therefore, it was decided to treat each continuous data segment as a complete DOT and reset t_0 and W_0 to match starting conditions at the beginning of each data segment. However, the values determined for K_i were forced to be the same across all data segments. The net effect of this approach was to determine one set of K_i for several "mini" DOT's. As can be seen in Figures 4.5-4.10, this approach produces significant improvement.

Using the improved method above for determining K_i, values for K_i for DOT's in each BPT on SFE#4 were used in ENERGY, a computer coded version of the presented method (Appendix D), to determine W_c and $E_{sm}=\sigma$. These results, as well as in-situ compressive rock strength ranges for both permeable and impermeable formations determined by Hareland [1992], are also given in Figures 4.5-4.10.

The ranges provided by Hareland's method illustrate how greatly in-situ compressive rock strength can fluctuate over small intervals (relative to the depth of the

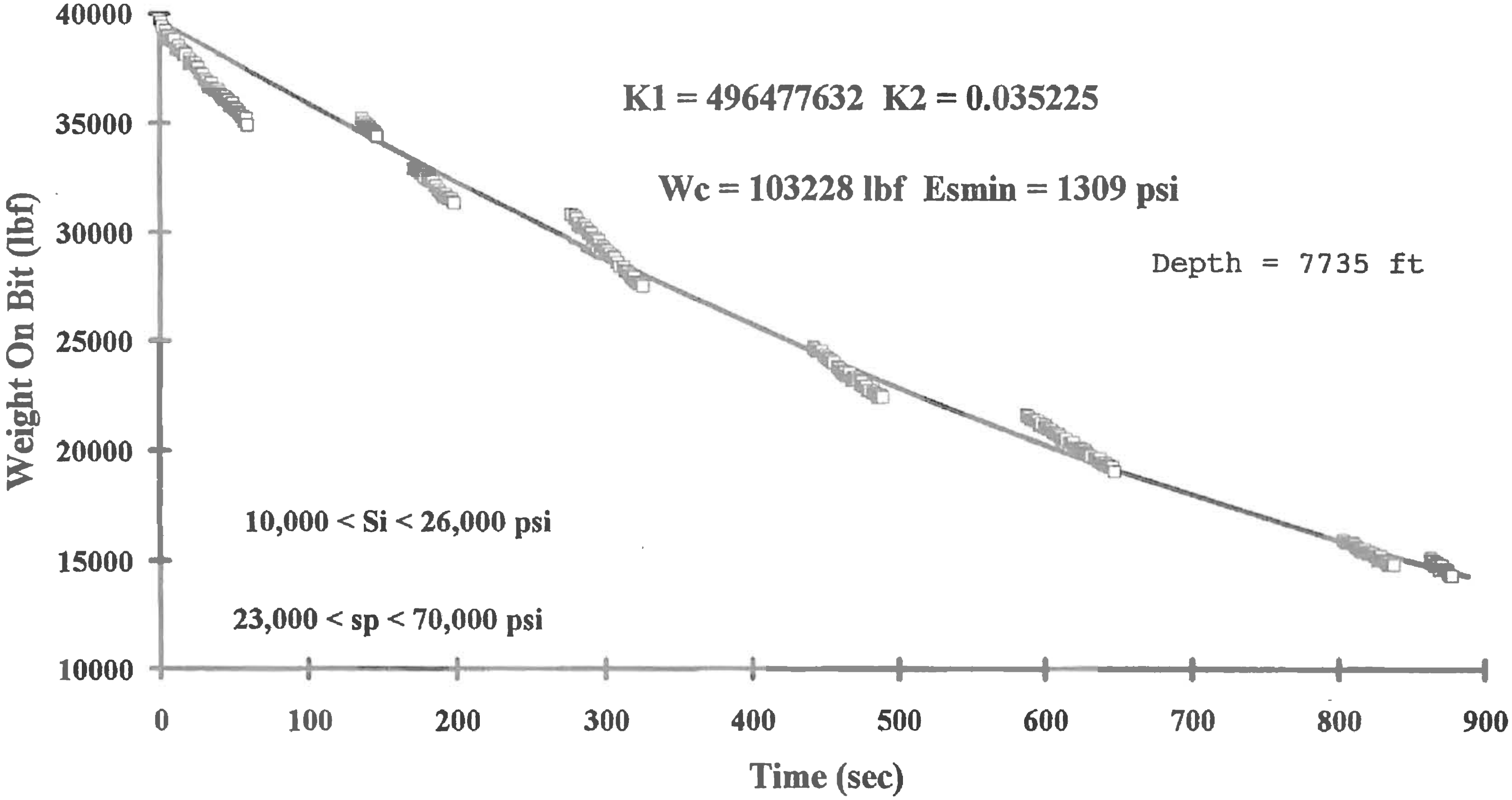


Figure 4.4 Curve Fit using BPT#8 as Continuous Drill-Off Test Data

TABLE 4.3

VALUES OF PARAMETERS USED IN (4.13) AND (4.14) FOR FIGURES 4.1 AND 4.4 - 4.8

τ	0.21
D	8.5 in.
E	30,000 psi
A	4.37 in.
L	Depth (given in figures)

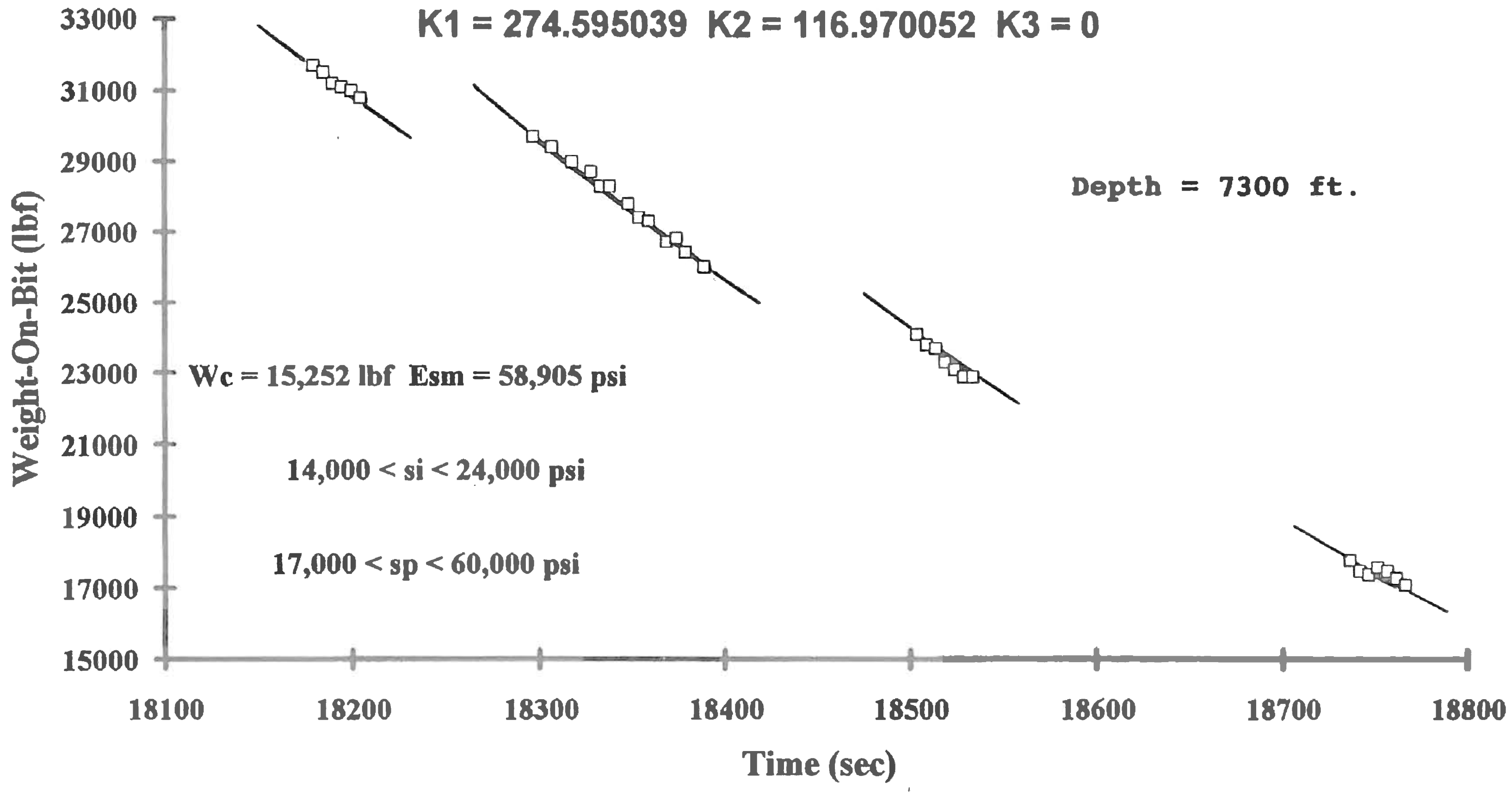


Figure 4.5 Curve Fit for SFE#4 - BPT#4 - 60 rpm

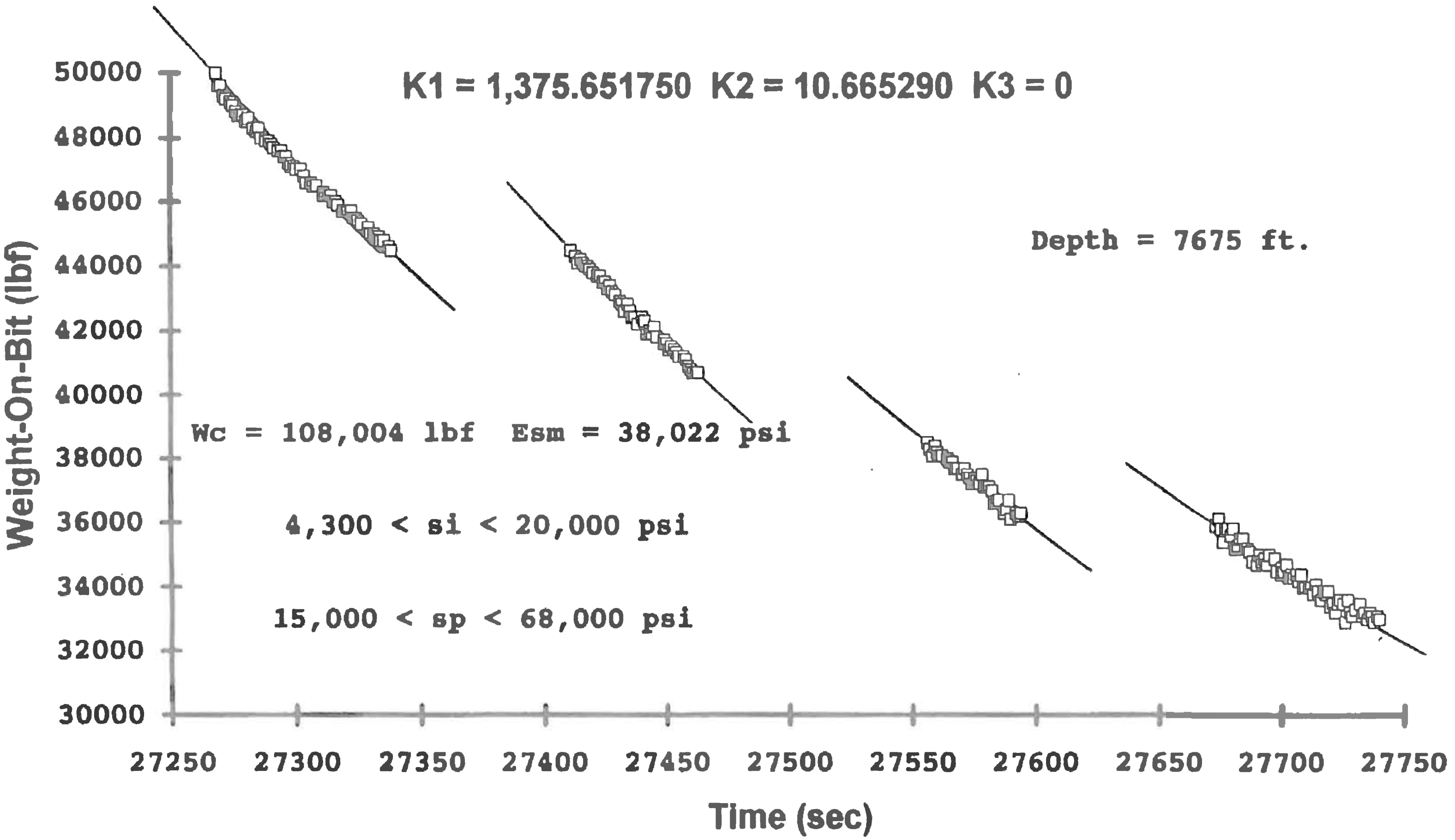


Figure 4.6 Curve Fit for SFE#4 - BPT#7 - 60 rpm

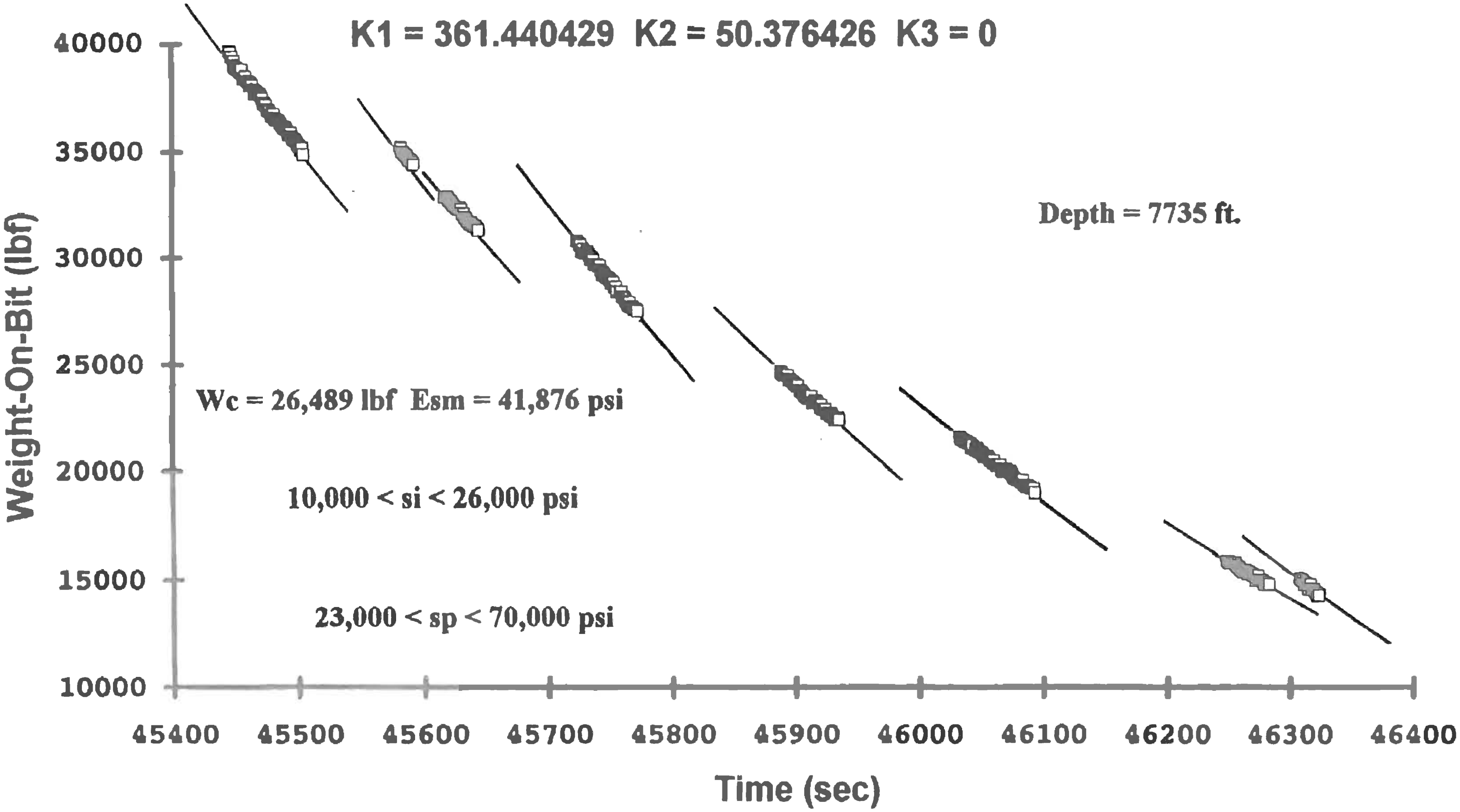


Figure 4.7 Curve Fit for SFE#4 -BPT#8 - 60 rpm

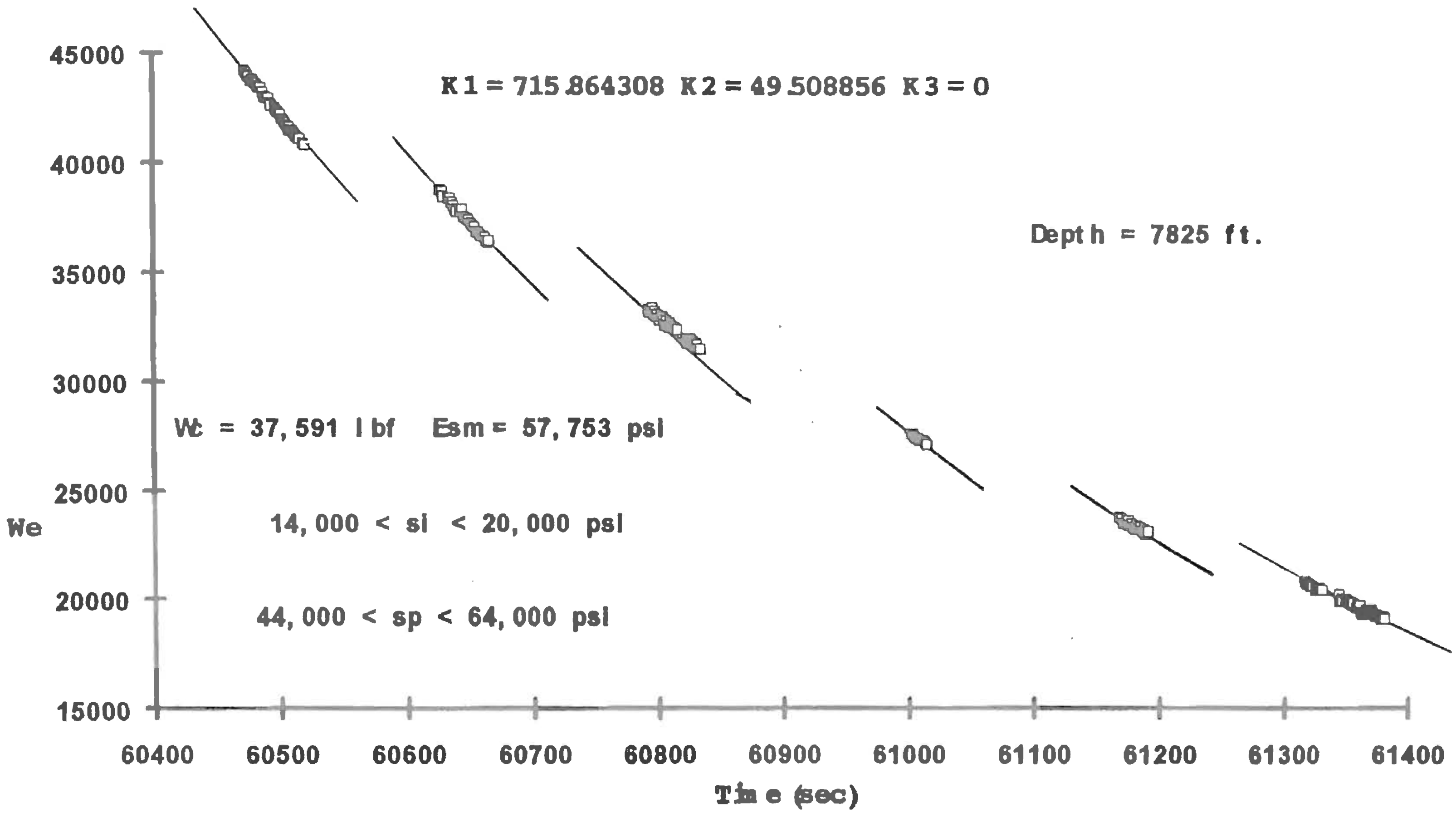


Figure 4.8 Curve Fit for SFE#4 - BPT#9 - 60 rpm

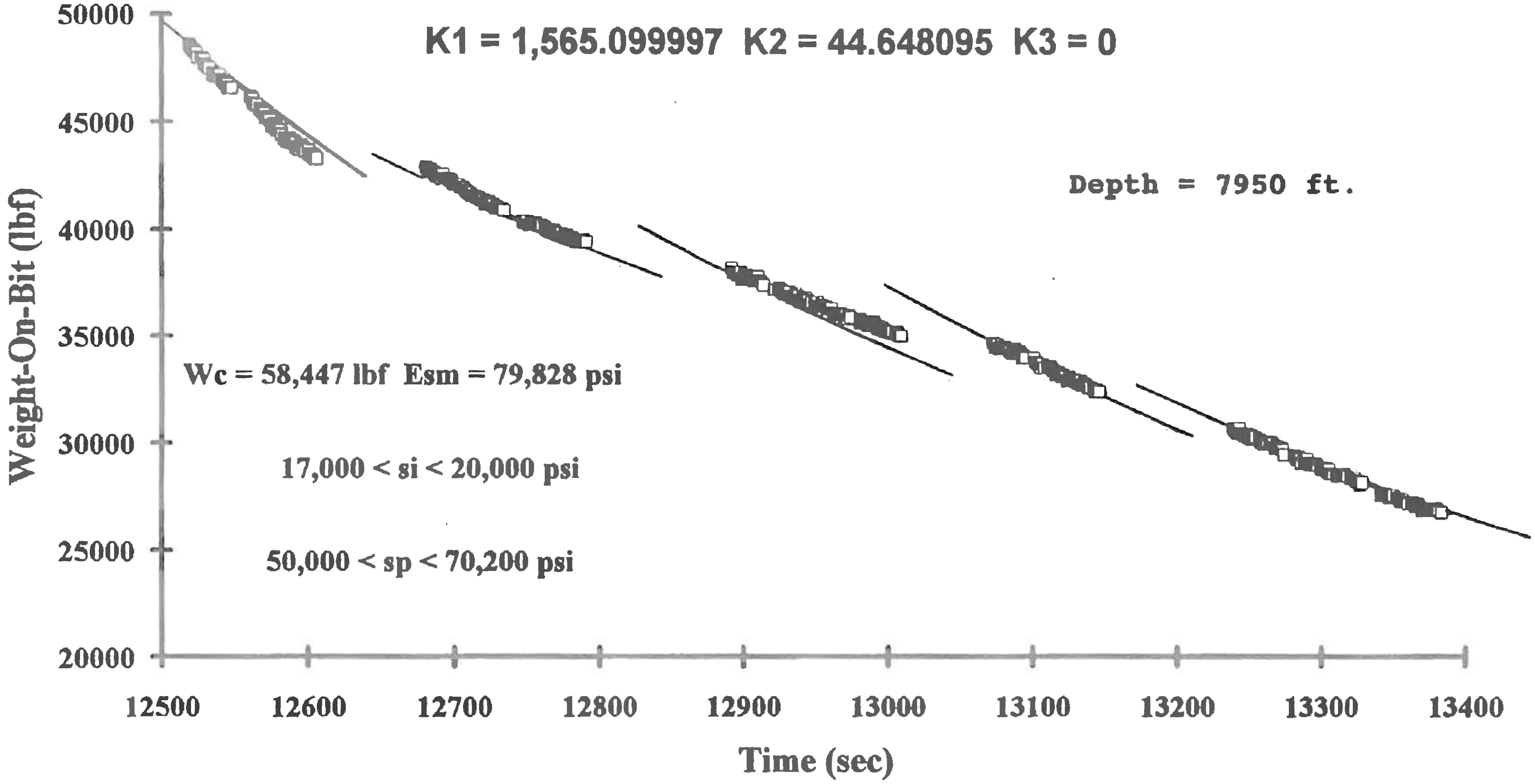


Figure 4.9 Curve Fit for SFE#4 - BPT#10 - 60 rpm

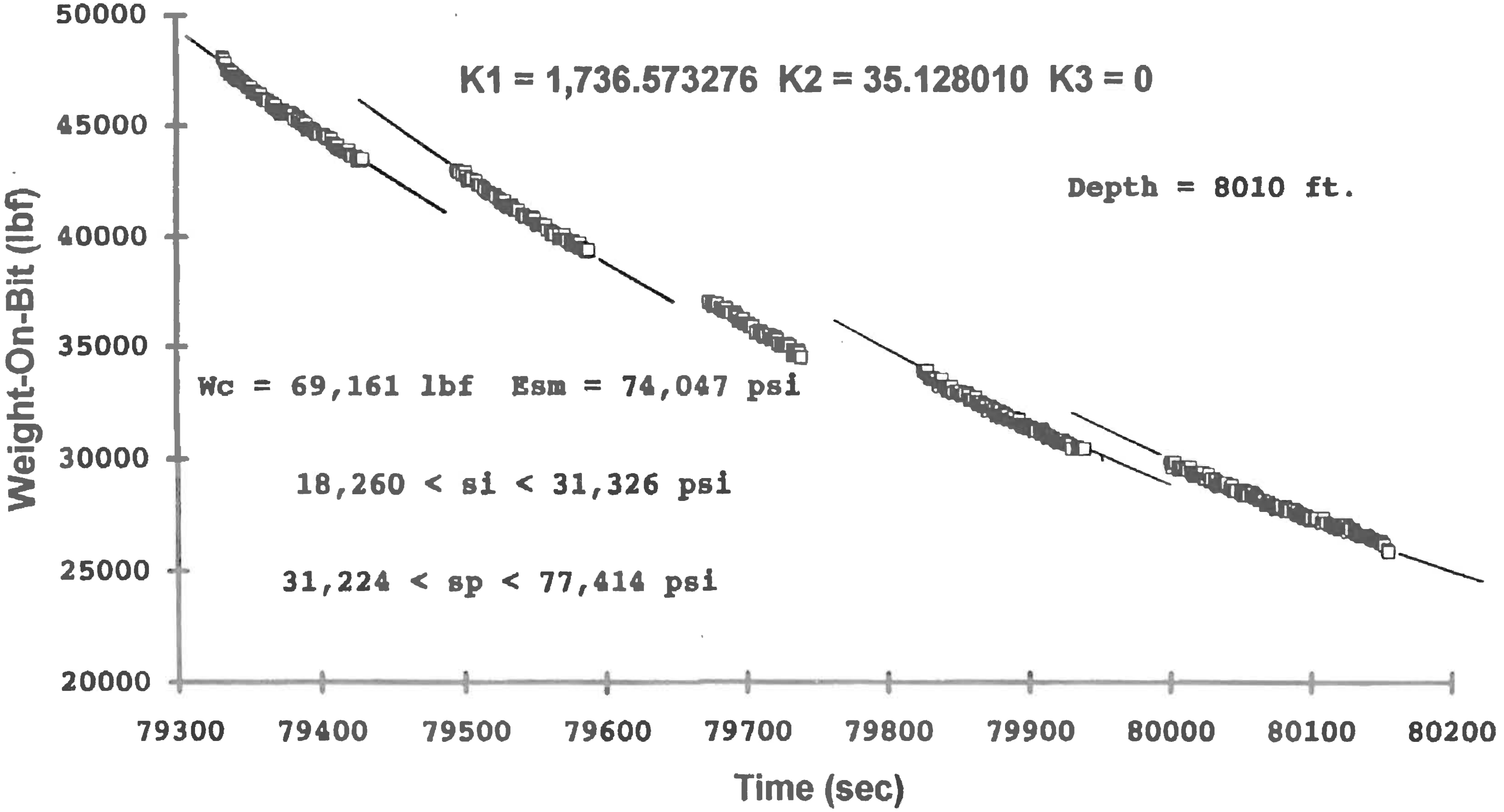


Figure 4.10 Curve Fit for SFE#4 - BPT#11 - 60 rpm

borehole). Both impermeable and permeable in-situ rock strength calculations can typically vary by more than 100 percent over a 20 foot interval. Hareland [1992] did not provide rock strength calculations at the exact depth of the DOT's, since the penetration rate could not be recorded during the DOT's.

In all cases the values calculated for E_{sm} are of the order of the confined rock strength as calculated by Hareland [1992], and in most cases within one of the stated ranges. Also, in all but one case, W_c is within an acceptable range for drilling. While these results are not conclusive, they do indicate that this new approach may be a simpler method of determining in-situ rock strength available and warrant further investigation, provided this method can be shown to be suitably robust. We examine robustness in what follows.

A sensitivity analysis was performed on the calculated specific energies and critical weight-on-bit values determined for the DOT's by varying K_1 and K_2 by +/- 20 percent about the calculated values. As shown in Table 4.4, a 20 percent change in K_1 and K_2 produces typically less than a ten percent variation in E_{sm} and W_c. This suggests that E_{sm} and W_c reflect the same level of accuracy as the drill-off test data, and that they are relatively insensitive to variations in K_1 and K_2. Thus, given reasonable continuous drill-off test data, the current method should return a reasonable approximation of in-situ rock strength, assuming good bottom-hole cleaning.

In this chapter, a reasonably robust method for determining in-situ rock strength has been developed. Considering the limited amount of acceptable data available, reasonable values for W_c and E_{sm} were calculated. This method does not require explicit knowledge of any rock properties, which are difficult to obtain. Also, this method does not have the inherent difficulties of determining chip hold-down effects or obtaining bit coefficients. All necessary quantities can be determined on site while drilling. This eliminates the need to know the condition of the bit, beyond knowing it is operable. This method deserves further study, using well behaved, continuous DOT data, in order to determine its true merit as a method to calculate rock strength.

TABLE 4.4

SENSITIVITY OF ENERGY AND CRITICAL WEIGHT-ON-BIT TO K_1 AND K_2

BPT	K_1	K_2	W_c, %error	E_{sm}, %error
4	219.68	116.97	-10.56	-10.56
	329.51	116.97	9.54	9.54
	274.60	93.58	11.68	-10.55
	274.60	140.36	-8.64	9.54
7	1100.52	10.67	-10.56	-10.56
	1650.78	10.67	9.54	9.54
	1375.65	8.54	10.49	-10.43
	1375.65	12.80	-7.98	9.45
8	289.15	50.38	-10.56	-10.56
	433.73	50.38	9.54	9.54
	361.44	40.30	11.50	-10.54
	361.44	60.46	-8.54	9.53
9	572.69	49.51	-10.56	-10.56
	859.03	49.51	9.54	9.54
	715.86	39.61	11.49	-10.54
	715.86	59.41	-8.54	9.53
10	1252.08	44.65	-10.58	-10.56
	1878.12	44.65	9.54	9.54
	1565.10	35.72	11.45	-10.54
	1565.10	53.58	-8.52	9.53
11	1389.26	35.13	-10.56	-10.56
	2083.88	35.13	9.54	9.54
	1736.57	28.10	11.35	-10.53
	1736.57	42.16	-8.47	9.52

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

Rock strength determination relies heavily on empiricism and difficult-to-obtain coefficients in Hareland and Hoberock's' [1993] method for determining in-situ stress bounds. This is due in no small part to the limited understanding of effective confining and pore pressure and their effects on drilling. This work has investigated three of Hareland and Hoberock's' assumptions in modeling these effects, and has proposed an alternative method to their rock strength calculation.

In Chapter II, we determined that for permeable formations, the difference between bottom hole and pore pressure, (P_b-P_p), is actually only an upper bound for the true dynamic differential pressure. A new quantity, formation fluid velocity, was developed to help determine when P_b-P_p is appropriate for use as the effective differential pressure. For formation fluid velocities on the order of the rate of bit penetration, true differential pressure drops to a minimum, approaching zero. However, as formation fluid velocity becomes increasingly small compared to penetration rate, the dynamic differential pressure approaches P_b-P_p. It was also determined that dynamic differential pressure varies from near zero to the maximum of P_b-P_p over most of the practical drilling conditions experienced in tight gas sands drilling.

In Chapter III, we examined static differential pressure for impermeable formations. Hareland and Hoberock calculate differential pressure for impermeable formations by setting differential pressure equal to bottom-hole pressure, P_b, which assumes pore pressure is zero near the bit. A theoretical study performed by Warren and Smith suggests that pore pressure might not reduce to zero and therefore, would invalidate this assumption. This possibility was tested with a single-element model of pore pressure reduction using data from the SFE#4 well, and the results were compared with Hareland and Hoberock's results. Their approach proved to return an in-situ stress bound calculation which either equaled or exceeded the known in-situ stresses for the well, while the approach employing the single element model did not. Therefore, it was concluded that Hareland and Hoberock's' approach, while empirical, is a better predictive tool for in-situ stress calculations in impermeable formations.

Hareland and Hoberock calculate static differential pressure for permeable formations as P_b-P_p, which assumes that the near bit pore pressure remains constant and equal to pore pressure remote to the borehole. This also assumes negligible communication between borehole and pore fluid due to the formation of a mud cake. For a mud cake to form, a certain amount of "spurt loss" of mud filtrate to the formation must occur. Given that the depth of interest is approximately one tooth penetration depth, spurt loss may be significant enough for pressure equalization across a drilled chip to occur. This would effectively reduce differential pressure to zero. Unfortunately, due to the inadequate modeling of differential pressure in the Hareland method, no means exists to adequately test this theory.

Perhaps the most significant contribution of this study is a new method for determining in-situ rock strength, which is simpler and potentially more accurate than the Hareland and Hoberock's approach. Described in Chapter IV, this method utilizes a common field test, the drill-off test, to collect data over a varying range of drilling parameters. This data is used to determine coefficients which are then used to calculate the minimum specific energy needed to drill a given formation. The minimum specific energy has been shown to approximate rock strength.

This method was tested using data collected during bit performance tests while drilling Staged Field Experiment Well #4. The results were compared with in-situ rock strengths determined using Hareland and Hoberock's' method. The minimum specific energies, in most cases, fell within the range of rock strengths given in Hareland [1992].

In order to determine if the minimum specific energy calculated using the approach developed in Chapter IV was sensitive to small variations in drill-off test data or calculation procedures, the coefficients determined from the bit performance tests were varied +/- 20 percent, and minimum specific energies were recalculated. The minimum specific energies were shown to vary approximately ten percent, suggesting that minimum specific energies calculated using the performance data are not unduly sensitive to data variations, and that the method is robust. While the results are not conclusive, the new approach appears to have merit and warrants further study.

The results of this work suggests several possible future studies:

1. The program DYNAMIC should be incorporated into PREDICT in order to determine the effect of dynamic differential pressure on permeable formation upper in-situ stress bound calculations.
2. More field tests of the new method for calculating in-situ rock strength should be performed in order to evaluate the method.
3. Torque data, calibrated according to the procedure given in Appendix D and measured while drilling, should be collected and used instead of the sliding coefficient of friction in the method for calculating in-situ compressive rock strength.
4. Procedures for automating the collection of and standards for the editing of drilling data needs to be developed.
5. Recommendations should be developed for frequency and depths of drill-off tests in applying the new method for calculating rock strengths.

REFERENCES

Bourgoyne, A.T., M.E. Chenevert, K.K. Millheim, and F.S. Young Jr. 1986. Applied Drilling Engineering, SPE Textbook Series, Vol. 2, SFE, Richardson, Tx.

Geertsma, J. 1957. The Effect of Fluid Pressure Decline on Volumetric Changes in Porous Rocks. J. Pet. Tech. 210:331-39. AIME, 210.

Hareland, G. 1992. Investigation of an In-Situ Stress Profile Model Using Drilling Parameters. Ph.D. Dissertation, Oklahoma State Univ., Stillwater, Okla.

Hareland, G. and L.L. Hoberock. 1993. Use of Drilling Parameters to Predict In-Situ Stress Bounds. SPE/IADC 25727. Presented at the 1993 SPE/IADC Drilling Conference in Amsterdam, The Netherlands, February 22-25, 1993.

Hoberock, L.L., G. Hareland, and G. Bratcher. 1992. Recommended Procedure to Calibrate Rig Torque in Ft-Lbs Using Rig Pipe Tongs, Internally Developed and Tested Procedure, Oklahoma State University, Stillwater, Oklahoma.

Holditch, S.A. and Associates. 1989. Application of Advanced Geological Retrophysical and Engineering Technologies to Evaluate and Improve Gas Recovery from Low Permeability Sandstone Reservoirs. Vol. 1. GRI Report. 89/0140. Gas Research Institute, Chicago, Ill.

Hubbert, M.K. 1969. The Theory of Ground-Water Motion and Related Papers. Hafner Publishing Company, Inc. New York, N.Y.

Hunt, E.E., E. Hunt Associates, Houston, Tx, Personal Communication, November 1991.

Hunt, E., G. Hareland, and L.L. Hoberock, 1992 Topical Report: Investigation of an In-Situ Stress Profile Model Using Drilling Parameters, GRI-92/0462, Gas Research Institute, Chicago, Ill.

Nobles, M.A. 1984. Using Computers to Solve Reservoir Engineering Problems, 2nd Ed. Gulf Publishing Co., Houston, Tx. p. 268.

Nur, A. and J.D. Byerlee. 1971. An Exact Effective Stress Law for Elastic Deformation of Rocks. Journal of Geophysical Research. 26:6414-19.

Peaceman, D.W. 1977. Fundamentals of Numerical Reservoir Simulation. Developments in Petroleum Science, No. 6. Elsevier Scientific Publishing Co., New York, N.Y., p. 7.

Pessier, R.C. and M.J. Fear 1992. Quantifying Common Drilling Problems with Mechanical Specific Energy and a Bit-Specific Coefficient of Sliding Friction. SPE 24584. 67th Annual Technical Conference and Exposition, Washington D.C. Oct 4-7

SAS Institute Inc., SAS/STAT Language, First Edition, Cary, N.C. Version 6.7 1992

Teale, R. 1965. The Concept of Specific Energy in Rock Drilling. Int. J. Rock Mech. Mining Sci. Vol. 2. Pergamon Press. pp. 57-73.

van der Knaap, W. 1959. Nonlinear Behavior of Porous Media. Trans., AIME 126, pp. 179-87.

Warren, T.M. 1987. Penetration Rate Performance of Roller Cone Bits, SPE Drilling Engineering, 9-18.

Warren, T.M. and M.B. Smith. 1985. Bottomhole Stress Factors Affecting Drilling Rate at Depth. J. Pet.Tech.. (August) 1523-33.

Winters, W.J., T.M. Warren, and E.C. Onyia. 1987. Roller Bit Model with Rock Ductility and Cone Offset, SPE 16696. 1987 SPE Annual Technical Conference, Houston, Tx.

APPENDIX A

COMPUTER PROGRAM: PREDICT

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                              c
c INSITU-STRESS AND ROCK STRENGTH PREDICTION PROGRAM c
c THIS PROGRAM ACCEPTS UP TO 1000 LINES OF DATA               c
c                                                              c
c     +------------------------------------------+             c
c     |      DEVELOPMENT INFORMATION       |                  c
c     +------------------------------------------+             c
c AUTHOR: G. BRATCHER, G. HARELAND, L. L. HOBEROCK           c
c                                                              c
c ORIGIN DATE:  1/28/92                                        c
c LANGUAGE: FORTRAN 77                                         c
c                                                              c
c     +------------------------------------+                   c
c     |      EXTERNAL ENVIRONMENT   |                          c
c     +------------------------------------+                   c
c ROUTINES CALLED:                                             c
c   DEMO    - DEMONSTRATES A TYPICAL ROCK STRENGTH             c
c             AND  INSITU-STRESS CALCULATION                   c
c   ACTUAL  - RUNS AN ACTUAL SESSION                           c
c   CALC    - PERFORMS THE ACTUAL STRENGTH AND STRESS          c
c           CALCULATIONS                                       c
c   BFILE   - WRITES A FILE CONTAINING DRILLING WIRE           c
c           INPUTS                                             c
c INPUT FILES:                                                 c
c   1  - DRIFIL = LITHOLOGY INPUTS                              c
c   2  - LOGFIL = DRILLING PARAMETER INPUTS                     c
c   5  - BITFIL = DRILLING WIRE INPUTS                          c
c   6  - PORFIL = PORE PRESSURE DATA                           c
c                                                              c
c OUTPUT FILES:                                                c
c   3  - ROKFIL = CONTAINS AT DEPTH ROCK STRENGTH               c
c   4  - STRFIL = CONTAINS AT DEPTH STRESS PROFILE              c
c                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c This program predicts rock strength and insitu-stress       c
c bounds from drilling parameters.                             c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c The array sizes in this program are set arbitrarily at       c
c 1000.  This makes it necessary to compile the program        c
c under the HUGE format. (fl /AH /Gt filename).  The array     c
c sizes may be changed, as long as they are consistent in      c
c size, to the users desire and available memory.  At the      c
c current setting, the program will run on a 1 MEG board.      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      IMPLICIT REAL(A-M)
      IMPLICIT REAL(O-Z)
      INTEGER M,N

      COMMON /MAT1/ N,LIME(1000),J1(1000),J2(1000),J3(1000),SAND(1000),
     $  SHALE(1000),DEPTH1(1000),PV(1000),ROP(1000),WOB(1000),
     $  RPM(1000),PP(1000),MW(1000),GPM(1000),BTYPE(1000),DIA(1000),
     $  COLDIA(1000),PIPDIA(1000),DRC(1000),FRIC(1000),YP(1000)
```

```
      CHARACTER*12 DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $     FRIFIL

      COMMON /MAT2/ DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $     FRIFIL

       COMMON /MAT3/ K(1000),KI(1000),SHI(1000),BHP(1000),
     $ BHPI(1000),ROCK(1000),ROCKI(1000),SH(1000),
     $ SOC(1000),SOCI(1000),SH2(1000),SHI2(1000),
     $ BETAP(1000),BETAI(1000)

 7     WRITE(*,*) '   1 - RUN INSITU STRESS DEMO PROGRAM '
      WRITE(*,*) '   2 - WRITE BIT FILE  '
      WRITE(*,*) '   3 - WRITE A PORE PRESSURE FILE '
      WRITE(*,*) '   4 - READ IN INITIAL FILES '
      WRITE(*,*) '   5 - RUN CALCULATION'
      WRITE(*,*) '   6 - CHANGE CURRENT VALUES IN A VARIABLE '
      WRITE(*,*) '       (This option should only be ran after '
      WRITE(*,*) '        Option 4) '
      WRITE(*,*) '   7 - SAVE CURRENT ARRAYS   '
      WRITE(*,*) '   8 - END SESSION (Remember to save the   '
      WRITE(*,*) '       working arrays and stress profiles,'
      WRITE(*,*) '       Option 7)'
      WRITE(*,*) ' ENTER THE FUNCTION YOU WOULD LIKE TO '
      WRITE(*,*) ' PERFORM '
      READ(*,*) M
      IF(M.EQ.1)THEN
         CALL DEMO()
         CALL CALC()
         CALL SAVE(2)
      ELSEIF(M.EQ.2)THEN
         CALL BFILE()
      ELSEIF(M.EQ.3)THEN
         CALL PORE()
      ELSEIF(M.EQ.4)THEN
         CALL ACTUAL()
      ELSEIF(M.EQ.5)THEN
         CALL CALC()
      ELSEIF(M.EQ.6)THEN
         CALL CHANGE()
      ELSEIF(M.EQ.7)THEN
         CALL SAVE(1)
      ELSEIF(M.EQ.8)THEN
         GOTO 6
      ENDIF
      GOTO 7

 6     STOP
      END
```

```
      SUBROUTINE DEMO()
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  This subroutine runs a simulation of a rock and insitu-          c
c  stress prediction, opening DEMO.BIT, DEMO.DRI, DEMO.POR,         c
c  and DEMO.LOG files needed to run the simulation, as well         c
c  as the output files DEMO.ROK and DEMO.STR output files.          c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      CHARACTER*12 DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $     FRIFIL

      COMMON /MAT2/ DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $      FRIFIL

      BITFIL='DEMO.BIT'
      DRIFIL='DEMO.DRI'
      LOGFIL='DEMO.LOG'
      PORFIL='DEMO.POR'
      ROKFIL='DEMO.ROK'
      STRFIL='DEMO.STR'
      CALL READ()
      RETURN
      END

      SUBROUTINE ACTUAL()
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  This subroutine drives the subroutine CALC.  ACTUAL allows       c
c  the user to enter the actual file names of the drilling          c
c  parameter files.  The names must be entered in single            c
c  quotes.                                                          c
c    Example:                                                       c
c        If the name of the lithology file is                       c
c        LITH.FIL and was on a disk in drive A:.  The               c
c        name will be entered:                                      c
c             'A:LITH.FIL'                                          c
c                                                                   c
c  The user must also enter two output file names, one for          c
c  the Rock Strength Output and one for the Insitu-Stress           c
c  Output.                                                          c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      CHARACTER*12 DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $    FRIFIL

      COMMON /MAT2/ DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $     FRIFIL

      WRITE(*,*)' ENTER THE NAME OF THE DRILLING DATA FILE IN '
      WRITE(*,*)' QUOTES (B:SFE2.LOG)'
      READ(*,*) LOGFIL
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*)' ENTER THE NAME OF THE LITHOLOGY FILE IN QUOTES '
      WRITE(*,*)' (B:SFE2.DRI) '
      READ(*,*) DRIFIL
      WRITE(*,*) ' '
```

```
      WRITE(*,*) ' '
      WRITE(*,*)' ENTER THE NAME OF THE BIT FILE IN QUOTES (B:SFE2.BIT)'
      READ(*,*)BITFIL
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*)' ENTER THE NAME OF THE PORE PRESSURE FILE (B:SFE2.POR)'
      READ(*,*) PORFIL
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      CALL READ()

      RETURN
      END

      SUBROUTINE CALC()
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c Subroutine CALC is the heart of the program.  It performs      c
c all of the necessary calculations for both rock strength       c
c and insitu-stress predictions.                                 c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c LIST OF VARIABLES READ FROM FILES                              c
c   N        - counter equal to the number of lines of           c
c              data to be read. THIS NUMBER SHOULD BE            c
c              WRITTEN AT THE BEGINNING OF THE LITHOLOGY         c
c              FILE, OTHERWISE THE PROGRAM WILL NOT RUN          c
c              PROPERLY                                          c
c   DEPTH1(I) - array containing the depth of the borehole       c
c              at which the drilling parameters are read         c
c   SHALE(I)  - array containing the percent of shale on         c
c              a foot by foot basis                              c
c   SILT      - percent of silt on a foot by foot basis.         c
c              Since the lithology coefficients (to be           c
c              discussed later) have not been determined         c
c              at this time, it is added to the percent          c
c              SAND and therefore, does not have to be           c
c              saved into an array.                              c
c   SAND(I)   - array containing the percent of sand on          c
c              a foot by foot basis                              c
c   CONGL     - percent of conglomerate on a foot by foot        c
c              basis.  As with SILT, the lithology               c
c              coefficients are undetermined, CONGL is           c
c              added to SAND.                                    c
c   LIME(I)   - array containing the percent of lime on          c
c              a foot by foot basis                              c
c   DOLO      - percent of dolomite on a foot by foot            c
c              basis.  The lithology coefficients have not       c
c              been determined for this formation, so it         c
c              is added to LIME.                                 c
c   COAL      - percent of coal on a foot by foot basis.         c
c              The lithology coefficients have not been          c
c              determined for this formation so it is            c
c              added to SHALE.                                   c
c   DEPTH2    - depth read from the parameter file.  Since       c
c              DEPTH1 was read and saved in an array,            c
```

```
c           DEPTH2 is redundant information and,               c
c           therefore, not saved.                              c
c   IDPH-                                                      c
c   DTC-                                                       c
c   DTS-                                                       c
c   RHOB-                                                      c
c   PEF-                                                       c
c   NPSS-                                                      c
c   ROP(I)    - array containing ROP (rate of penetration)     c
c           in feet per hour on a foot by foot basis           c
c   TOR      - reads torque from the drilling parameter        c
c           file.  This is a dummy parameter since             c
c           torque is not used in any calculations.            c
c   WOB(I)   - array containing WOB (weight-on-bit) in         c
c           klb on a foot by foot basis                        c
c   RPM(I)    - array containing RPM (rotary speed) in         c
c           revolutions per minute                             c
c   SPM      - strokes per minute.  Since this parameter       c
c           is not used in calculations, it is not             c
c           saved in an array.                                 c
c   DEPTH3    - depth read from the bit file.  Since           c
c           DEPTH1 was read and saved in an array,             c
c           DEPTH3 is redundant information and,               c
c           therefore, not saved.                              c
c   BTYPE(I) - array containing the IADC bit code at           c
c           DEPTH3 (=DEPTH1)                                   c
c   DIA(I)    - array containing the bit diameter              c
c   J1(I),                                                     c
c   J2(I),                                                     c
c   J3(I)     - array containing jet diameters in 1/32 IN.     c
c           Ex. if jet diameter = 11/32, J3(I)=11              c
c   MW(I)     - array containing the mud weight at DEPTH3      c
c           (LB/GAL)                                           c
c   PV(I)     - array containing the plastic viscosity of      c
c           the drilling mud at DEPTH3 (CENTIPOISE)            c
c   GPM(I)    - array containing flow rate (GAL/MIN)           c
c   COLDIA(I) - array containing collar diameters at given     c
c           depths (INCHES)                                    c
c   PIPDIA(I) - array containing pipe diameter at given        c
c           depth (INCHES)                                     c
c   DRC(I)    - array containing the length of the drill       c
c           collars at a given depth (FEET)                    c
c                                                              c
c CONSTANTS USED IN CALCULATIONS                               c
c   A       - bit design constant dependent on the IADC        c
c           bit code (HR RPM IN / FT)                          c
c   B       - bit design constant dependent on the IADC        c
c           bit code (HR RPM IN / FT)                          c
c   C       - bit design constant dependent on the IADC        c
c           bit code (HR LBF GAL / FT LB CP IN)                c
c VARIABLES CALCULATED IN THE PROGRAM                          c
c   PP(I)    - array containing pore pressure (PSI)            c
c   TERM1     - first term in the Three-Term Bit Model         c
c   TERM1I    - first term in the Three-Term Bit Model         c
```

```
c           Impermeable Case                                  c
c  TERM2    - second term in the Three-Term Bit Model         c
c  TERM3    - third term in the Three-Term Bit Model          c
c  TERM3I   - third term in the Three-Term Bit Model          c
c           Impermeable Case                                  c
c  AREAJ    - total area of the bit jets (SQUARE INCHES)      c
c  AREAB    - bit area (SQUARE INCHES)                        c
c  IMFORC   - impact force (LBF)                              c
c  ALPHA    - constant used in calculating MODIFIED           c
c           IMPACT FORCE                                      c
c  MIFORCE  - modified impact force (LBF)                     c
c  BHPI(I)  - bottom hole pressure, Impermeable Case          c
c           (PSI)                                             c
c  BHPI(I)  - bottom hole pressure, Permeable Case            c
c           (PSI)                                             c
c  KLII     - K coefficient for lime, Impermeable Case        c
c  KSHI     - K coefficient for shale, Impermeable Case       c
c  KSAI     - K coefficient for sand, Impermeable Case        c
c  KLI      - K coefficient for lime, Permeable Case          c
c  KSH      - K coefficient for shale, Permeable Case         c
c  KSA      - K coefficient for sand, Permeable Case          c
c  KI(I)    - K coefficient for complete formation,           c
c           Impermeable Case                                  c
c  K(I)     - K coefficient for complete formation,           c
c           Permeable Case                                    c
c  ROCKYI   - rock strength squared, Impermeable Case         c
c  ROCKI(I) - rock strength, Impermeable Case (PSI)           c
c  ROCKY    - rock strength squared, Permeable Case           c
c  ROCK(I)  - rock strength, Permeable Case (PSI)             c
c  SAND1P-                                                    c
c  SAND1I-                                                    c
c  LIME1P-                                                    c
c  LIME1I-                                                    c
c  SHAL1P-                                                    c
c  SHAL1I-                                                    c
c  SAND2P-                                                    c
c  SAND2I-                                                    c
c  LIME2P-                                                    c
c  LIME2I-                                                    c
c  SHAL2P-                                                    c
c  SHAL2I-                                                    c
c  SOC(I) - unconfined rock strength, Permeable Case          c
c           (PSI)                                             c
c  SOCI(I)  - unconfined rock strength, Permeable Case        c
c           (PSI)                                             c
c  CPRP     - confining pressure, Permeable Case (PSI)        c
c  CPRI     - confining pressure, Impermeable Case (PSI)      c
c  GAMMAP,                                                    c
c  ZETAP    - offsets of confining pressure, Permeable        c
c           Case (PSI)                                        c
c  GAMMAI,                                                    c
c  ZETAI    - offsets of confining pressure, Impermeable      c
c           Case (PSI)                                        c
c  SC1P-                                                      c
```

```
c   SC1I-                                                     c
c   SC2P-                                                     c
c   SC2I-                                                     c
c   DELTAP-                                                   c
c   DELTAI-                                                   c
c   BETAP    - angle of internal friction, Permeable Case     c
c            (DEGREE)                                         c
c   BETAI    - angle of internal friction, Impermeable        c
c            Case (DEGREE)                                    c
c   KO-                                                       c
c   KOI      - , Impermeable Case                             c
c   SOVB     - overburden stress (PSI)                        c
c   SH(I)    - horizontal stress, Permeable Case (PSI)        c
c   SHI(I)   - horizontal stress, Impermeable Case (PSI)      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      IMPLICIT REAL(A-Z)
      INTEGER N,I
      REAL NP,NI,KLII,KSHI,KSAI,KLI,KO,KOI,KSH,KSA,IMFORC,MIFORC,
     $ LIME1,LIME2,LIME1P,LIME1I,LIME2P,LIME2I,AFL,BFL,CFL,
     $ AFSH,BFSH,CFSH,AFSA,BFSA,CFSA,A1,B1,A2,B2,A3,B3,CPDEL

      CHARACTER*12 BITFIL,DRIFIL,LOGFIL,ROKFIL,STRFIL,
     $  PORFIL,FRIFIL

      COMMON /MAT1/ N,LIME(1000),J1(1000),J2(1000),J3(1000),SAND(1000),
     $ SHALE(1000),DEPTH1(1000),PV(1000),ROP(1000),WOB(1000),
     $ RPM(1000),PP(1000),MW(1000),GPM(1000),BTYPE(1000),DIA(1000),
     $ COLDIA(1000),PIPDIA(1000),DRC(1000),FRIC(1000),YP(1000)

      COMMON /MAT2/ DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $    FRIFIL

      COMMON /MAT3/ K(1000),KI(1000),SHI(1000),BHP(1000),
     $ BHPI(1000),ROCK(1000),ROCKI(1000),SH(1000),
     $ SOC(1000),SOCI(1000),SH2(1000),SHI2(1000),
     $ BETAP(1000),BETAI(1000)

      DATA AFL,BFL,CFL/.01413,.4702,.6595/,
     $ AFSH,BFSH,CFSH/.00496,.7572,.1025/,
     $  AFSA,BFSA,CFSA/.01413,.4702,.6595/,
     $  A1,B1,A2,B2,A3,B3/.01331,.57106,.0043188,.74191,
     $ .0043188,.74191/

      CPDEL=50.
C ---- BEGINNING OF FOOT BY FOOT CALCULATIONS
C  N = NUMBER OF DATA POINTS
      DO 20 I=1,N
      WRITE(*,*) I
C ---- CHOOSING PROPER BIT COEFFICIENTS
      IF(BTYPE(I).EQ.437)THEN
        A=.01817
        B=3.0709
        C=.002094
```

```
      ELSEIF(BTYPE(I).EQ.517)THEN
        A=.02587
        B=4.2149
        C=.00335
      ELSEIF(BTYPE(I).EQ.527)THEN
C  THESE ARE NOT THE REAL VALUES, AT THE TIME OF THIS PROGRAM
C  THE ACTUAL COEFFICIENTS WERE NOT AVAILABLE
        A=0.01383
        B=9.77070
        C=0.002231
      ELSEIF(BTYPE(I).EQ.537)THEN
        A=.01383
        B=9.7704
        C=.002231
      ELSEIF(BTYPE(I).EQ.617)THEN
        A=.01902
        B=13.4527
        C=.003256
      ELSEIF(BTYPE(I).EQ.627)THEN
        A=.01953
        B=3.2536
        C=.01441
      ELSEIF(BTYPE(I).EQ.737)THEN
        A=.03224
        B=9.314
        C=.007988
      ENDIF

C ----- INTERMEDIATE CALCUALTIONS
      TERM2=(B*WOB(I)**2.)/(A*DIA(I)**4.)
      AREAJ=.000767*(J1(I)**2.+J2(I)**2.+J3(I)**2.)
      AREAB=3.14593*(DIA(I)**2)/4.
      IMFORC=.000516*MW(I)*GPM(I)*(.32086*GPM(I)/AREAJ)
      ALPHA=.15*AREAB/AREAJ
      MIFORC=IMFORC*(1.-ALPHA**(-.122))


C ----- IMPERMEABLE ROCK STRENGTH CALCULATIONS
C  FRIC(I)=ANTOT

      BHPI(I)=.052*MW(I)*DEPTH1(I)+FRIC(I)
      IF(BHPI(I).LT.120.)BHPI(I)=120.
      KLII=CFL+AFL*(BHPI(I)-120.)**BFL
      KSHI=CFSH+AFSH*(BHPI(I)-120.)**BFSH
      KSAI=CFSA+AFSA*(BHPI(I)-120.)**BFSA
      KI(I)=LIME(I)*KLII+SHALE(I)*KSHI+SAND(I)*KSAI
      TERM1I=(RPM(I)*WOB(I)**2.)/(A*KI(I)*ROP(I)*DIA(I)**3.)
      TERM3I=(C*RPM(I)*MW(I)*PV(I)*WOB(I)**2.)/
     $   (MIFORC*KI(I)*A*DIA(I)**2.)
      ROCKYI=TERM1I-TERM2-TERM3I
      IF(ROCKYI.LT.0.5)THEN
        ROCKI(I)=3.
      ELSE
        ROCKI(I)=(TERM1I-TERM2-TERM3I)**.5
```

```
      ENDIF
      ROCKI(I)=ROCKI(I)*1000.

C ----- PERMEABLE ROCK STRENGTH CALCULATIONS
C   FRIC(I) = ANTOT

      BHP(I)=.052*MW(I)*DEPTH1(I)-PP(I)+FRIC(I)
      IF(BHP(I).LT.120.)BHP(I)=120.
      KLI=CFL+AFL*(BHP(I)-120.)**BFL
      KSH=CFSH+AFSH*(BHP(I)-120.)**BFSH
      KSA=CFSA+AFSA*(BHP(I)-120.)**BFSA
      K(I)=LIME(I)*KLI+SHALE(I)*KSH+SAND(I)*KSA
      TERM1=(RPM(I)*WOB(I)**2.)/(A*K(I)*ROP(I)*DIA(I)**3.)
      TERM3=(C*RPM(I)*MW(I)*PV(I)*WOB(I)**2.)/(MIFORC*K(I)*A*DIA(I)**2)
      ROCKY=TERM1-TERM2-TERM3
      IF(ROCKY.LT.0.5)THEN
         ROCK(I)=3.
      ELSE
         ROCK(I)=(TERM1-TERM2-TERM3)**.5
      ENDIF
      ROCK(I)=ROCK(I)*1000.
20    CONTINUE

C ------ INITIAL GUESS AT INSITU-STRESS
      DO 40 I=1,N
      SAND1=SAND(I)*(1.+A1*BHP(I)**B1)
      LIME1=LIME(I)*(1.+A2*BHP(I)**B2)
      SHALE1=SHALE(I)*(1.+A3*BHP(I)**B3)
      SAND2=SAND(I)*(1.+A1*BHPI(I)**B1)
      LIME2=LIME(I)*(1.+A2*BHPI(I)**B2)
      SHALE2=SHALE(I)*(1.+A3*BHPI(I)**B3)
      SOC(I)=ROCK(I)/(SAND1+SHALE1+LIME1)
      SOCI(I)=ROCKI(I)/(SAND2+SHALE2+LIME2)
      CPR=.65*DEPTH1(I)-PP(I)
      GAMMA=CPR+CPDEL
      ZETA=CPR-CPDEL
      SAND1=SAND(I)*(1.+A1*ZETA**B1)
      LIME1=LIME(I)*(1.+A2*ZETA**B2)
      SHALE1=SHALE(I)*(1.+A3*ZETA**B3)
      SAND2=SAND(I)*(1.+A1*GAMMA**B1)
      LIME2=LIME(I)*(1.+A2*GAMMA**B2)
      SHALE2=SHALE(I)*(1.+A3*GAMMA**B3)
      SC1=SOC(I)*(SAND1+LIME1+SHALE1)
      SC1I=SOCI(I)*(SAND1+SHALE1+LIME1)
      SC2=SOC(I)*(SAND2+LIME2+SHALE2)
      SC2I=SOCI(I)*(SAND2+SHALE2+LIME2)
      DELTA=SC2-SC1
      DELTAI=SC2I-SC1I
      DELTA=DELTA/(DELTA+4.*CPDEL)
      DELTAI=DELTAI/(DELTAI+4.*CPDEL)
      BETA1=ASIN(DELTA)*57.29578
      BETAI1=ASIN(DELTAI)*57.29578
      KO=1.-DELTA
      KOI=0.9*(1.-DELTA)
```

```
      SOVB=1.04*DEPTH1(I)
      SH(I)=KO*(SOVB-PP(I))+PP(I)
      SHI(I)=KOI*(SOVB-PP(I))+PP(I)
40    CONTINUE

C ----- BEGINNING OF INSITU-STRESS ITERATION
      DO 50 I=1,N
      DO 60 J=1,8
      CPRP=SH(I)-PP(I)
      CPRI=SHI(I)-PP(I)
      GAMMAP=CPRP+CPDEL
      GAMMAI=CPRI+CPDEL
      ZETAP=CPRP-CPDEL
      ZETAI=CPRI-CPDEL
      SAND1P=SAND(I)*(1.+A1*ZETAP**B1)
      SAND1I=SAND(I)*(1.+A1*ZETAI**B1)
      LIME1P=LIME(I)*(1.+A2*ZETAP**B2)
      LIME1I=LIME(I)*(1.+A2*ZETAI**B2)
      SHAL1P=SHALE(I)*(1.+A3*ZETAP**B3)
      SHAL1I=SHALE(I)*(1.+A3*ZETAI**B3)
      SC1P=SOC(I)*(SAND1P+LIME1P+SHAL1P)
      SC1I=SOCI(I)*(SAND1I+LIME1I+SHAL1I)
      SAND2P=SAND(I)*(1.+A1*GAMMAP**B1)
      SAND2I=SAND(I)*(1.+A1*GAMMAI**B1)
      LIME2P=LIME(I)*(1.+A2*GAMMAP**B2)
      LIME2I=LIME(I)*(1.+A2*GAMMAI**B2)
      SHAL2P=SHALE(I)*(1.+A3*GAMMAP**B3)
      SHAL2I=SHALE(I)*(1.+A3*GAMMAI**B3)
      SC2P=SOC(I)*(SAND2P+LIME2P+SHAL2P)
      SC2I=SOCI(I)*(SAND2I+LIME2I+SHAL2I)
      DELTAP=SC2P-SC1P
      DELTAI=SC2I-SC1I
      NP=DELTAP/(DELTAP+4.*CPDEL)
      NI=DELTAI/(DELTAI+4.*CPDEL)
      BETAP(I)=ASIN(NP)*57.29578
      BETAI(I)=ASIN(NI)*57.29578
      KO=1.-NP
      KOI=.9*(1.-NI)
      SOVB=1.04*DEPTH1(I)
      SH2(I)=SH(I)
      SHI2(I)=SHI(I)
      SH(I)=KO*(SOVB-PP(I))+PP(I)
      SHI(I)=KOI*(SOVB-PP(I))+PP(I)
      IF(ABS(SH(I)-SH2(I)).LT..01.AND.
     $   ABS(SHI(I)-SHI2(I)).LT..01)GOTO 50
60    CONTINUE
50    CONTINUE

1002  RETURN
      END

      SUBROUTINE BFILE()
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  SUBROUTINE BFILE provides a faster way to enter data from        c
```

```
c  the Drilling Wires than a foot by foot entry of data.              c
c  Data is entered over intervals.  The interval is defined           c
c  as the distance over which no parameter changes.  The data         c
c  to be entered into a file in this subroutine is:                   c
c                                                                     c
c    BITFIL    - dummy file name that stores an entered               c
c              name                                                   c
c    TOTAL     - maximum depth of file (FT)                           c
c    DEPTH     - initial depth of the file (FT)                       c
c    DEPIN(I)  - depth at which BTYPE(I) goes in the hole             c
c              or when another parameter changes (FT)                 c
c    BTYPE(I)  - IADC bit code number                                 c
c    DEPOUT(I) - depth at which BTYPE(I) comes out of the             c
c              hole or another parameter changes (FT)                 c
c    MW(I)     - mud weight (LB / GAL)                                c
c    PV(I)     - plastic viscosity (CENTIPOISE)                       c
c    GPM(I)    - flow rate (GAL / MIN)                                c
c    DIAJET    - jet diameter (INCHES)                                c
c    DIA(I)    - bit diameter (INCHES)                                c
c    COLDIA(I) - drilling collar diameter (INCHES)                    c
c    PIPDIA(I) - pipe diameter (INCHES)                               c
c    DRC(I)    - drilling collar diameter (INCHES)                    c
c    M,JMAX,J  - counters                                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                     c
c    YP(I)     - drilling mud yield point ( lbs/100 sq.ft.)           c
c                                                                     c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      IMPLICIT REAL(A-Z)
      REAL DEPTH1,DEPTH
      INTEGER DEPIN,DEPOUT,JMAX,J,JMIN,M1
      CHARACTER*12 BITFIL

      WRITE(*,*) ' This routine writes a file which contains '
      WRITE(*,*) ' information taken from the Drilling Wires. '
      WRITE(*,*) ' '
      WRITE(*,*) ' You will need to know: the interval of recording '
      WRITE(*,*) ' (in feet) of the lithology file, the initial and '
      WRITE(*,*) ' final depths of the hole to be studied in feet, '
      WRITE(*,*) ' the IADC bit code, the depths at which these bits'
      WRITE(*,*) ' were pulled in feet, the weight of the drilling '
      WRITE(*,*) ' mud for each bit interval in lb/gal, the plastic '
      WRITE(*,*) ' viscocity for this interval in Centipoise, the '
      WRITE(*,*) ' yield point of the drilling mud in lb/100 sg.ft.,'
      WRITE(*,*) ' the mud flow rate in GPM for each interval, the '
      WRITE(*,*) ' diameters of each jet, bit, the largest collar,'
      WRITE(*,*) ' and the drill string in inches, and the length'
      WRITE(*,*) ' of the drill collars in feet.'
      WRITE(*,*) ' '
      WRITE(*,*) ' The output file will contain all of this information'
      WRITE(*,*) ' plus the pressure loss due to friction of the '
      WRITE(*,*) ' drilling mud'
      WRITE(*,*) ''
      WRITE(*,*) ' '
```

```
    WRITE(*,*) ' Enter the Bit, Mud, and Hydralics File Name in'
    WRITE(*,*) ' single quotes'
    READ(*,*)BITFIL
    WRITE(*,*) ' '
    WRITE(*,*) ' '
    OPEN(UNIT=5,FILE=BITFIL,STATUS='NEW')

    WRITE(*,*) ' '
    WRITE(*,*) ' '
    WRITE(*,*) ' Enter the footage interval size of recording used '
    WRITE(*,*) ' in the lithology file. Ex. If the data is taken '
    WRITE(*,*) ' every 1/2 foot, enter 0.5. '
    READ(*,*) STEP

    WRITE(*,*) ' Enter the final depth of the hole (ft). '
    READ(*,*) TOTAL
    WRITE(*,*) ' '
    WRITE(*,*) ' '
    WRITE(*,*) ' Enter the depth, in feet, at which you wish to start'
    WRITE(*,*) ' entering data '
    READ(*,*) DEPTH

    JMIN=1
60   IF(DEPTH.GE.TOTAL)THEN
       GOTO 80
    ELSE
       DEPIN=DEPTH
       WRITE(*,*) ' '
       WRITE(*,*) ' '
       WRITE(*,2000) DEPTH
       READ(*,*) BTYPE
       WRITE(*,*) ' '
       WRITE(*,*) ' '
       WRITE(*,*) ' Enter the depth, in feet, at which this bit was '
       WRITE(*,*) ' pulled. This must be a whole number. '
       READ(*,*) DEPOUT
       WRITE(*,*) ' '
       WRITE(*,*) ' '
       IF(DEPOUT.GT.TOTAL)DEPOUT=TOTAL
       WRITE(*,*) ' Enter the Mud Weight, in lb/gal,'
       WRITE(*,*) ' for this interval.'
       READ(*,*) MW
       WRITE(*,*) ' '
       WRITE(*,*) ' '
       WRITE(*,*) ' Enter the Plastic Viscosity, in Centipoise,'
       WRITE(*,*) ' for the drilling mud for this interval.'
       READ(*,*) PV
       WRITE(*,*) ' '
       WRITE(*,*) ' '
       WRITE(*,*) ' Enter the Yield Point, in lb/100 sq.ft.,'
       WRITE(*,*) ' of the drilling mud for this interval.'
       READ(*,*) YP
       WRITE(*,*) ' '
       WRITE(*,*) ' '
```

```
      WRITE(*,*) ' Enter the Mud Flow Rate, in gpm, for '
      WRITE(*,*) ' this interval '
      READ(*,*) GPM
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' If all of the jet diameters are equal'
      WRITE(*,*) ' enter 1; otherwise enter 2'
      READ(*,*) M1
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      IF(M1.EQ.1)THEN
        WRITE(*,*) ' Enter the diameter for the jets in'
        WRITE(*,*) ' 1/32 inches '
        READ(*,*) J1
        J2=J1
        J3=J1
        WRITE(*,*) ' '
        WRITE(*,*) ' '
      ELSE
        WRITE(*,*) ' Enter the diameter for jet 1 in '
        WRITE(*,*) ' 1/32 inches'
        READ(*,*) J1
        WRITE(*,*) ' '
        WRITE(*,*) ' '
        WRITE(*,*) ' Enter the diameter for jet 2 in 1/32 inches'
        READ(*,*) J2
        WRITE(*,*) ' '
        WRITE(*,*) ' '
        WRITE(*,*) ' Enter the diameter for jet 3 in 1/32 inches'
        READ(*,*) J3
        WRITE(*,*) ' '
        WRITE(*,*) ' '
      ENDIF
      WRITE(*,*) ' Enter the Bit Diameter in decimal inches '
      READ(*,*) DIA
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' Enter the diameter of the largest drill collar  '
      WRITE(*,*) ' in decimal inches.'
      READ(*,*) COLDIA
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' Enter the total length of the drill collars'
      WRITE(*,*) ' in feet.'
      READ(*,*) DRC
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' Enter the outside Pipe Diameter in decimal'
      WRITE(*,*) ' inches.'
      READ(*,*) PIPDIA
      DEPTH=DEPOUT
c
c  WRITING THE DRILLING WIRE INPUT FILE
c
```

```
      JMAX=JMAX+(DEPOUT-DEPIN)/STEP
      IF(DEPOUT.GE.TOTAL)JMAX=JMAX+1

C---- FINDING THE FRICTION LOSS FOR THE BHA AND 1 FT. OF PIPE
      CALL HYDRA(PRESSC,PRESSP,GPM,MW,PV,YP,COLDIA,DIA,PIPDIA,DRC)
C      FRIC=PRESSC+PRESSP*(DEPIN-1.)
C
C      CORRECTION MADE JULY 21, 1992
C      MUST ACCOUNT FOR DRILL COLLAR LENGTH
C
      FRIC=PRESSC+PRESSP*(DEPIN-1.-DRC)

      DEPTH1=DEPIN-STEP

      DO 100 J=JMIN,JMAX
      FRIC=FRIC+PRESSP*STEP*(J-1)
      DEPTH1=DEPTH1+STEP
      WRITE(5,2001)DEPTH1,BTYPE,DIA,J1,J2,J3,
     $   MW,PV,GPM,COLDIA,PIPDIA,DRC,FRIC,YP
100   CONTINUE
      JMIN=JMAX+1

      ENDIF
      GOTO 60

 80   CONTINUE
      CLOSE(5)

2000  FORMAT(1X,' AT DEPTH =',F9.1,
     $ ' ENTER THE IADC BIT CODE (3-digits)')
2001  FORMAT(1X,14F8.2)

      RETURN
      END



      SUBROUTINE PORE()
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                 c
c  This subroutine writes a pore pressure file to be read by      c
c  the subroutine CALC.                                           c
c                                                                 c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      REAL DEPIN,DEPOUT,DEPTH,DELPP,PG,PP,DELGRD

      INTEGER J,JMAX,JMIN,M
      CHARACTER*12 PPFIL,PGFIL

      WRITE(*,*) ' SUBROUTINE PORE (pressure) provides an'
      WRITE(*,*) ' easy way to enter pore pressure for the'
      WRITE(*,*) ' drilling interval to be studied. '
      WRITE(*,*) ' '
      WRITE(*,*) ' Enter the name of the Pore Pressure File '
```

```
      WRITE(*,*) ' to be created '
      READ(*,*) PPFIL
      OPEN(UNIT=9,FILE=PPFIL,STATUS='NEW')
      WRITE(*,*) ' '
      WRITE(*,*) ' Enter the name of the pressure gradient file'
      READ(*,*) PGFIL
      OPEN(UNIT=10,FILE=PGFIL,STATUS='NEW')
      WRITE(*,*) ''
      WRITE(*,*) ''
      WRITE(*,*) ' Enter the starting depth of the hole'
      WRITE(*,*) ' to be studied (ft).'
      READ(*,*) DEPIN
      WRITE(*,*) ''
      WRITE(*,*) ' Enter the final depth of the hole to '
      WRITE(*,*) ' be studied (ft). '
      READ(*,*) DEPTH2

      JMIN=1
3000  IF(DEPIN.GE.DEPTH2)THEN
       GOTO 3005
      ELSE
       WRITE(*,*) ' Enter the Pore Pressure Gradient in PSI/ft for'
       WRITE(*,*) ' the interval starting at depth ',DEPIN,' ft.'
       READ(*,*) PG
       WRITE(*,*) ' '
       WRITE(*,*) ' Enter the depth at which this pressure '
       WRITE(*,*) ' gradient interval ends (ft).'
       READ(*,*) DEPOUT
       IF(DEPOUT.GT.DEPTH2)DEPOUT=DEPTH2
       WRITE(*,*) ' '
       WRITE(*,*) ' If this is a constant Pressure Gradiant over'
       WRITE(*,*) ' the interval enter a 2, otherwise enter a 1 '
       READ(*,*) M
       WRITE(*,*) ' '
       IF(M.EQ.1)THEN
        WRITE(*,*)' Enter the change in pore pressure gradient'
        WRITE(*,*)' (PSI/ft/ft)'
        READ(*,*) DELGRD
       ELSE
        DELGRD=0.0
       ENDIF
        WRITE(*,*)' If there is a step change in pore pressure '
        WRITE(*,*)' at the beginning of this interval, enter this'
        WRITE(*,*)' change in PSI, otherwise enter zero (0.0).'
        WRITE(*,*)' (Be sure the sign of the step change reflects'
        WRITE(*,*)' the direction of the step change).'
        READ(*,*) DELPP
        WRITE(*,*) ''
        WRITE(*,*) ''
      WRITE(10,3006) DEPIN,DEPOUT,PG,DELGRD,DELPP
3006  FORMAT(2X,F10.3,2X,F10.2,2X,F5.3,2X,F10.7,2X,F10.3)
      ENDIF
```

```
      WRITE(*,*) ' Enter the footage increment used in the '
      WRITE(*,*) ' lithology or drilling parameter file. '
      WRITE(*,*) ' '
      WRITE(*,*) ' Ex. If the depth column appears as:'
      WRITE(*,*) '    '
      WRITE(*,*) '    7001.5 '
      WRITE(*,*) '    7002.0 '
      WRITE(*,*) '    7002.5 '
      WRITE(*,*) '    7003.0 '
      WRITE(*,*) ''
      WRITE(*,*) ' Enter the number 0.5 '
      READ(*,*) STEP

      DEPTH4=DEPIN - STEP
      JMAX=JMAX+(DEPOUT-DEPIN)/STEP
      IF(DEPOUT.EQ.DEPTH2)JMAX=JMAX+1
      DO 3002 J=JMIN,JMAX
       DEPTH4=DEPTH4+STEP
       DEPTH=DEPTH4
       DELDEP=DEPTH4-DEPIN
       PP=DEPTH4*(PG+DELGRD*DELDEP)+DELPP
       WRITE(9,3003)DEPTH,PP
       WRITE(*,3003)DEPTH,PP
3003   FORMAT(1X,F10.2,5X,F12.3)
3002  CONTINUE
      JMIN=JMAX+1
      DEPIN=DEPOUT
      GOTO 3000

3005  CONTINUE
       CLOSE(9)
       CLOSE(10)
       RETURN
       END

       SUBROUTINE HYDRA(PRESSC,PRESSP,GPM,MW,PV,
     $ YP,COLDIA,DIA,PIPDIA,DRC)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                   c
c  This subroutine calculates friction loss with a simplified      c
c  HYDRAlics model which is then read into SUBROUTINE CALC.        c
c                                                                   c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       IMPLICIT REAL(A-Z)

         ANCVEL=GPM/(2.448*(DIA**2.-COLDIA**2.0))
         ANPVEL=GPM/(2.448*(DIA**2.-PIPDIA**2.0))
         FAN3=PV+YP
         FAN6=FAN3+PV
         PLN=3.32*LOG(FAN6/FAN3)
         PLK=(510.*FAN3)/(511.)**PLN
         REYAC=109000.*MW*((ANCVEL)**(2.-PLN))/PLK
         REYBC=(.0208*(DIA-COLDIA)/(2.+1./PLN))**PLN
         REYNC=REYAC*REYBC
```

```
       REYAP=109000*MW*((ANPVEL)**(2.-PLN))/PLK
       REYBP=(.0208*(DIA-PIPDIA)/(2+1/PLN))**PLN
       REYNP=REYAP*REYBP

     IF(REYNC.GT.2100.1)THEN
        FRICKC=.0791/REYNC**.25
        PRESSC=DRC*(FRICKC*MW*ANCVEL**2.)/
    $    (21.1*(DIA-COLDIA))
     ELSE
        PRESSCA=DRC*(PLK*ANCVEL**PLN)
        PRESSCB=((2+1/PLN)/.0208)**PLN
        PRESSCC=144000.*(DIA-COLDIA)**(1-PLN)
        PRESSC=PRESSCA*PRESSCB/PRESSCC
      ENDIF

C   PRESSURE LOSS IS NOT MULTIPLIED BY THE LENGTH OF THE PIPE
C   SINCE A ONE FOOT LENGTH IS ASSUMED.  THE TOTAL PRESSURE LOSS
C   DUE TO THE LENGTH OF THE PIPE IS CALCULATED IN BITFIL.
     IF(REYNP.GT.2100.1)THEN
      FRICKP=0.0791/REYNP**.25
      PRESSP=(FRICKP*MW*ANPVEL**2.)/
    $ (21.1*(DIA-PIPDIA))
     ELSE
      PRESSPA=(PLK*ANPVEL**PLN)
      PRESSPB=((2+1/PLN)/.0208)**PLN
      PRESSPC=144000.*(DIA-PIPDIA)**(1-PLN)
      PRESSP=PRESSPA*PRESSPB/PRESSPC
     ENDIF

     RETURN
     END


     SUBROUTINE CHANGE()
C---  THIS ROUTINE ALLOWS THE USER TO EDIT THE EXISTING ARRAYS
C---  IN ORDER TO RUN DRILLING SIMULATIONS

     IMPLICIT REAL(A-Z)
     INTEGER M,M1,M2,M3,L,N

     CHARACTER*12 BITFIL,DRIFIL,LOGFIL,ROKFIL,STRFIL,
    $   PORFIL,FRIFIL

     COMMON /MAT2/ DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
    $     FRIFIL

     COMMON /MAT1/ N,LIME(1000),J1(1000),J2(1000),J3(1000),SAND(1000),
    $ SHALE(1000),DEPTH1(1000),PV(1000),ROP(1000),WOB(1000),
    $ RPM(1000),PP(1000),MW(1000),GPM(1000),BTYPE(1000),DIA(1000),
    $ COLDIA(1000),PIPDIA(1000),DRC(1000),FRIC(1000),YP(1000)

     WRITE(*,*)' Enter the number of the variable you wish to change.'
     WRITE(*,*)' '
```

```
      WRITE(*,*)' 1 - Percent lithology '
      WRITE(*,*)' 2 - Rate of Penetration (ft/hr) '
      WRITE(*,*)' 3 - Weight on Bit (klbs)'
      WRITE(*,*)' 4 - Rotary Speed (rmp)'
      WRITE(*,*)' 5 - Bit Type (IADC)'
      WRITE(*,*)' 6 - Bit Diameter (in.) '
      WRITE(*,*)' 7 - Jet Diameter (1/32in.)'
      WRITE(*,*)' 8 - Pipe Diameter (inches)'
      WRITE(*,*)' 9 - Collar Diameter (in.)'
      WRITE(*,*)' 10- Mud Weight (lbs/gal) '
      WRITE(*,*)' 11- Plastic Visc. (cps)'
      WRITE(*,*)' 12- Mud Flow Rate (GPM) '
      WRITE(*,*)' 13- Collar Length (ft)'
      WRITE(*,*)' 14- Yield Point (lb/100 sq. yrd.)'
      WRITE(*,*)' 15- EXIT TO MAIN '

      READ(*,*) M

      IF(M.EQ.15)GOTO 4030

      WRITE(*,*)' If you want to change the parameter over an interval'
      WRITE(*,*)' enter a 1; if you want to change the parameter'
      WRITE(*,*)' foot-by-foot enter a 2. '
      WRITE(*,*)' '
      READ(*,*) M1

      IF(M.EQ.1)THEN
C -----
        IF(M1.EQ.1)THEN
           CALL LTHCHG(DEPTH1,LIME,SHALE,SAND)
        ELSE
           CALL FTLCHG(DEPTH1,LIME,SHALE,SAND)
        ENDIF
C -----
      ELSEIF(M.EQ.2)THEN
C -----
        IF(M1.EQ.1)THEN
          CALL CHANG2(DEPTH1,ROP)
        ELSE
          CALL FTBYFT(DEPTH1,ROP)
        ENDIF
C -----
      ELSEIF(M.EQ.3)THEN
C -----
        IF(M1.EQ.1)THEN
          CALL CHANG2(DEPTH1,WOB)
        ELSE
          CALL FTBYFT(DEPTH1,WOB)
        ENDIF
C -----
      ELSEIF(M.EQ.4)THEN
C -----
        IF(M1.EQ.1)THEN
          CALL CHANG2(DEPTH1,RPM)
```

```
      ELSE
        CALL FTBYFT(DEPTH1,RPM)
      ENDIF
C -----
    ELSEIF(M.EQ.5)THEN
C -----
      IF(M1.EQ.1)THEN
        CALL CHANG2(DEPTH1,BTYPE)
      ELSE
        CALL FTBYFT(DEPTH1,BTYPE)
      ENDIF
C -----
    ELSEIF(M.EQ.6)THEN
C -----
      IF(M1.EQ.1)THEN
        CALL CHANG2(DEPTH1,DIA)
      ELSE
        CALL FTBYFT(DEPTH1,DIA)
      ENDIF
      CALL NUFRIC()
C -----
    ELSEIF(M.EQ.7)THEN
C -----
    WRITE(*,*)' If you would like all three jets to be equal in '
    WRITE(*,*)' diameter, enter 1, otherwise enter 2 '
    READ(*,*) M2
      IF(M2.EQ.1)THEN
       IF(M1.EQ.1)THEN
        CALL CHANG2(DEPTH1,J1)
       ELSE
        CALL FTBYFT(DEPTH1,J1)
       ENDIF
C -----
      DO 4035 L=1,1000
        J2(L)=J1(L)
        J3(L)=J1(L)
4035      CONTINUE
      ELSEIF(M2.EQ.2)THEN
       WRITE(*,*)' Enter the number of the jet with the diameter '
       WRITE(*,*)' you wish to change (1,2,or3).'
       READ(*,*) M3
       IF(M3.EQ.1)THEN
        IF(M1.EQ.1)THEN
         CALL CHANG2(DEPTH1,J1)
        ELSE
         CALL FTBYFT(DEPTH1,J1)
        ENDIF
       ELSEIF(M3.EQ.2)THEN
        IF(M1.EQ.1)THEN
         CALL CHANG2(DEPTH1,J2)
        ELSE
         CALL FTBYFT(DEPTH1,J2)
        ENDIF
       ELSEIF(M3.EQ.3)THEN
```

```
            IF(M1.EQ.1)THEN
              CALL CHANG2(DEPTH1,J3)
            ELSE
              CALL FTBYFT(DEPTH1,J3)
            ENDIF
          ENDIF
        ENDIF
C -----
      ELSEIF(M.EQ.8)THEN
         IF(M1.EQ.1)THEN
           CALL CHANG2(DEPTH1,PIPDIA)
         ELSE
           CALL FTBYFT(DEPTH1,PIPDIA)
         ENDIF
         CALL NUFRIC()
C -----
      ELSEIF(M.EQ.9)THEN
C -----
        IF(M1.EQ.1)THEN
           CALL CHANG2(DEPTH1,COLDIA)
        ELSE
           CALL FTBYFT(DEPTH1,COLDIA)
        ENDIF
        CALL NUFRIC()
C -----
      ELSEIF(M.EQ.10)THEN
C -----
        IF(M1.EQ.1)THEN
           CALL CHANG2(DEPTH1,MW)
        ELSE
           CALL FTBYFT(DEPTH1,MW)
        ENDIF
        CALL NUFRIC()
C -----
      ELSEIF(M.EQ.11)THEN
C -----
        IF(M1.EQ.1)THEN
           CALL CHANG2(DEPTH1,PV)
        ELSE
           CALL FTBYFT(DEPTH1,PV)
        ENDIF
        CALL NUFRIC()
C -----
      ELSEIF(M.EQ.12)THEN
C -----
        IF(M1.EQ.1)THEN
            CALL CHANG2(DEPTH1,GPM)
        ELSE
            CALL FTBYFT(DEPTH1,GPM)
        ENDIF
        CALL NUFRIC()
C -----
      ELSEIF(M.EQ.13)THEN
C -----
```

```
      IF(M1.EQ.1)THEN
        CALL CHANG2(DEPTH1,DRC)
      ELSE
        CALL FTBYFT(DEPTH1,DRC)
      ENDIF
      CALL NUFRIC()
C ——
     ELSEIF(M.EQ.14)THEN
C ——
      IF(M1.EQ.1)THEN
       CALL CHANG2(DEPTH1,YP)
      ELSE
       CALL FTBYFT(DEPTH1,YP)
      ENDIF
      CALL NUFRIC()
     ENDIF

4030 CONTINUE
     CLOSE(20)
     RETURN
     END

     SUBROUTINE NUFRIC()
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C RECULATES FRICTIONAL LOSSES AFTER ONE OF THE PARAMETERS    C
C   AFFECTING FRICTIONAL LOSSES IS CHANGED                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     IMPLICIT REAL(A-Z)
     INTEGER N,J

     COMMON /MAT1/ N,LIME(1000),J1(1000),J2(1000),J3(1000),SAND(1000),
    $  SHALE(1000),DEPTH1(1000),PV(1000),ROP(1000),WOB(1000),
    $  RPM(1000),PP(1000),MW(1000),GPM(1000),BTYPE(1000),DIA(1000),
    $  COLDIA(1000),PIPDIA(1000),DRC(1000),FRIC(1000),YP(1000)

C --- FINDING THE FRICTION LOSS FOR THE BHA AND 1 FT. OF PIPE
     DO 100 J=1,N
     CALL HYDRA(PRESSC,PRESSP,GPM,MW,PV,YP,COLDIA,DIA,PIPDIA,DRC)
     FRIC(J)=PRESSC+PRESSP*DEPTH1(J)
100  CONTINUE

     RETURN
     END

     SUBROUTINE READ()
C—— THIS SUBROUTINE READS IN THE NECESSARY DATA FOR
C—— CALCULATIONS FROM THE APPROPRIATE FILE
     IMPLICIT REAL(A-Z)

     REAL IDPH,DTC,DTS,RHOB,PEF,NPSS

     INTEGER I,N

     CHARACTER*12 BITFIL,DRIFIL,LOGFIL,ROKFIL,STRFIL,
```

```
     $  PORFIL,FRIFIL

      COMMON /MAT1/ N,LIME(1000),J1(1000),J2(1000),J3(1000),SAND(1000),
     $ SHALE(1000),DEPTH1(1000),PV(1000),ROP(1000),WOB(1000),
     $ RPM(1000),PP(1000),MW(1000),GPM(1000),BTYPE(1000),DIA(1000),
     $ COLDIA(1000),PIPDIA(1000),DRC(1000),FRIC(1000),YP(1000)

      COMMON /MAT2/ DRIFIL,LOGFIL,ROKFIL,STRFIL,BITFIL,PORFIL,
     $     FRIFIL


      OPEN(UNIT=1,FILE=DRIFIL,STATUS='OLD')
      OPEN(UNIT=2,FILE=LOGFIL,STATUS='OLD')
      OPEN(UNIT=5,FILE=BITFIL,STATUS='OLD')
      OPEN(UNIT=6,FILE=PORFIL,STATUS='OLD')

      READ(1,*)N
      DO 10 I=1,N
      READ(1,*) DEPTH1(I),SHALE(I),SILT,SAND(I),CONGL,
     $  LIME(I),DOLO,COAL
      SAND(I)=SAND(I)+SILT+CONGL
      SHALE(I)=SHALE(I)+COAL
      LIME(I)=LIME(I)+DOLO
      READ(2,*) DEPTH2,IDPH,DTC,DTS,RHOB,PEF,
     $  NPSS,ROP(I),TOR,WOB(I),RPM(I),SPM
      IF(DEPTH2.NE.DEPTH1(I))THEN
         WRITE(*,1003) DRIFIL,LOGFIL
         GOTO 1002
      ENDIF
      READ(5,*)DEPTH3,BTYPE(I),DIA(I),J1(I),J2(I),J3(I),MW(I),
     $ PV(I),GPM(I),COLDIA(I),PIPDIA(I),DRC(I),FRIC(I),YP(I)
      IF(DEPTH3.NE.DEPTH1(I))THEN
         WRITE(*,1003) DRIFIL,BITFIL
         GOTO 1002
      ENDIF
      READ(6,*) DEPTH4,PP(I)
      IF(DEPTH4.NE.DEPTH1(I))THEN
         WRITE(*,1003) DRIFIL,PORFIL
         GOTO 1002
      ENDIF
10    CONTINUE

1003  FORMAT(' THE DEPTHS FROM FILE ',A12,' AND ',
     $   A12,' DO NOT MATCH! THE FILES ARE NOT COMPATIBLE',
     $   ' AND MUST BE EDITED.')

1002  CONTINUE
       CLOSE(1)
       CLOSE(2)
       CLOSE(5)
       CLOSE(6)
       RETURN
       END
```

```
      SUBROUTINE LTHCHG(DEPTH1,DUM1,DUM2,DUM3)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL(A-Z)
      REAL DEPTH1(1000),DUM1(1000),DUM2(1000),DUM3(1000)
      INTEGER I

      WRITE(*,*)' Enter the beginning depth, in feet, of the interval'
      WRITE(*,*)' to be changed.'
      READ(*,*) DEPSTRT
      WRITE(*,*)' Enter the end depth, in feet, of the interval to be'
      WRITE(*,*)' changed.'
      READ(*,*) DEPEND
      WRITE(*,*)' Enter the new lithology values in decimal form '
      WRITE(*,*)' Ex. If Shale makes up 10% of the formation, enter'
      WRITE(*,*)' 0.10. (Enter no more than 2 decimal places.)'
      WRITE(*,*)'  '
      WRITE(*,*)'  '
      WRITE(*,*)' LIME '
      READ(*,*) LIME
      WRITE(*,*)'  '
      WRITE(*,*)'  '
      WRITE(*,*)' SHALE '
      READ(*,*) SHALE
      WRITE(*,*)'  '
      WRITE(*,*)'  '
      WRITE(*,*)' SAND '
      READ(*,*) SAND

      DO 4029 I=1,1000
      IF(DEPTH1(I).GE.DEPSTRT.AND.DEPTH1(I).LE.DEPEND)THEN
       DUM1(I)=LIME
       DUM2(I)=SHALE
       DUM3(I)=SAND
      ENDIF
4029  CONTINUE

      RETURN
      END

      SUBROUTINE CHANG2(DEPTH1,DUM1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL(A-Z)
      REAL DEPTH1(1000),DUM1(1000),DUM2(1000)
      INTEGER I
      DATA DUM2/1000*0/

      DO 4040 I=1,1000
4040   DUM2(I)=DUM1(I)

      WRITE(*,*)' Enter the beginning depth, in feet, of the interval'
      WRITE(*,*)' to be changed.'
      READ(*,*) DEPSTRT
```

```
      WRITE(*,*)' Enter the end depth, in feet, of the interval to be'
      WRITE(*,*)' changed.'
      READ(*,*) DEPEND
      WRITE(*,*)' ENTER THE NEW VALUE OF THE VARIABLE '
      READ(*,*) VALUE

      DO 4029 I=1,1000
      IF(DEPTH1(I).GE.DEPSTRT.AND.DEPTH1(I).LE.DEPEND)THEN
        DUM2(I)=VALUE
      ENDIF
4029  CONTINUE

      DO 4041 I=1,1000
4041  DUM1(I)=DUM2(I)
      RETURN
      END

      SUBROUTINE FTLCHG(DEPTH1,DUM1,DUM2,DUM3)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL(A-Z)
      REAL DEPTH1(1000),DUM1(1000),DUM2(1000),DUM3(1000)
      INTEGER I,J,N

      DO 4031 I=1,1000
      WRITE(*,*)' Enter the depth, in feet, you wish to change.'
      READ(*,*) DEPTH
      WRITE(*,*)' Ex. If Shale makes up 10% of the formation, enter'
      WRITE(*,*)' 0.10. (Enter no more than 2 decimal places.)'
      WRITE(*,*)'  '
      WRITE(*,*)'  '
      WRITE(*,*)' LIME '
      READ(*,*) LIME
      WRITE(*,*)'  '
      WRITE(*,*)'  '
      WRITE(*,*)' SHALE '
      READ(*,*) SHALE
      WRITE(*,*)'  '
      WRITE(*,*)'  '
      WRITE(*,*)' SAND '
      READ(*,*) SAND

      DO 4032 J=1,1000
      IF(DEPTH1(J).EQ.DEPTH)THEN
        DUM1(J)=LIME
        DUM2(J)=SHALE
        DUM3(J)=SAND
      ENDIF
4032  CONTINUE
      WRITE(*,*)' CONTINUE (1), END (0)'
      READ(*,*) N

      IF(N.EQ.0)GOTO 4034
4031  CONTINUE
```

```
4034  CONTINUE
      RETURN
      END

      SUBROUTINE FTBYFT(DEPTH1,DUM1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL(A-Z)
      REAL DEPTH1(1000),DUM1(1000),DUM2(1000)
      INTEGER I,J,N
      DATA DUM2/1000*0/

      DO 4042 I=1,1000
4042  DUM2(I)=DUM1(I)

      DO 4031 I=1,1000
      WRITE(*,*)' ENTER THE DEPTH OF THE CHANGE'
      READ(*,*) DEPTH
      WRITE(*,*)' ENTER THE NEW VALUE '
      READ(*,*) VALUE

      DO 4032 J=1,1000
      IF(DEPTH1(J).EQ.DEPTH)THEN
        DUM2(J)=VALUE
          GOTO 4033
      ENDIF
4032  CONTINUE
4033  CONTINUE
      WRITE(*,*)' CONTINUE (1), END (0)'
      READ(*,*) N

      IF(N.EQ.0)GOTO 4034
4031  CONTINUE
4034  CONTINUE
      DO 4043 I=1,1000
4043  DUM1(I)=DUM2(I)
      RETURN
      END

       SUBROUTINE SAVE(CODE)
C -- THIS ROUTINE SAVES THE CURRENT WORKING ARRAYS IN USER
C -- DEFINED OUTPUT FILES IN A FORMAT WHICH CAN BE RELOADED
C -- INTO THE PROGRAM AT A LATER DATE
      IMPLICIT REAL(A-Z)
      INTEGER N,I,CODE
      CHARACTER*12 LOGBAK,DRIBAK,BITBAK,ROKFIL,STRFIL
C       CHARACTER*12 PORBAK
      COMMON /MAT1/ N,LIME(1000),J1(1000),J2(1000),J3(1000),SAND(1000),
     $  SHALE(1000),DEPTH1(1000),PV(1000),ROP(1000),WOB(1000),
     $  RPM(1000),PP(1000),MW(1000),GPM(1000),BTYPE(1000),DIA(1000),
     $  COLDIA(1000),PIPDIA(1000),DRC(1000),FRIC(1000),YP(1000)

       COMMON /MAT3/ K(1000),KI(1000),SHI(1000),BHP(1000),
     $  BHPI(1000),ROCK(1000),ROCKI(1000),SH(1000),
```

```
     $ SOC(1000),SOCI(1000),SH2(1000),SHI2(1000),
     $ BETAP(1000),BETAI(1000)

      IF(CODE.EQ.1)THEN

      WRITE(*,*)' Enter the name of the Drilling Data OUTPUT File in '
      WRITE(*,*)' quotes.  This file will contain the final drilling'
      WRITE(*,*)' parameters (Ex. ROP,WOB,etc.) (B:SFE2.LOG)'
      READ(*,*) LOGBAK
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*)' Enter the name of the Lithology OUTPUT File in'
      WRITE(*,*)' quotes.  This file will contain the final lithology '
      WRITE(*,*)' parameters (Ex. %sand,%shale,etc)(B:SFE2.DRI) '
      READ(*,*) DRIBAK
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*)' Enter the name of the Bit, Mud, and Hydralics OUTPUT'
      WRITE(*,*)' File in quotes. This file will contain the final '
      WRITE(*,*)' drilling data collected from the drilling wires'
      WRITE(*,*)' (Ex. bit type,collar length,etc)(B:SFE2.BIT)'
      READ(*,*)BITBAK
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*)' Enter the name of the Rock Strength OUTPUT '
      WRITE(*,*)' File in quotes. This file will contain the last rock '
      WRITE(*,*)' strengths calculated.'
      READ(*,*)ROKFIL
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*)' Enter the name of the In-Situ Stress OUTPUT File'
      WRITE(*,*)' in quotes.  This file will contain the final in- '
      WRITE(*,*)' situ stress calculations.'
      READ(*,*)STRFIL

      OPEN(UNIT=1,FILE=DRIBAK,STATUS='NEW')
      OPEN(UNIT=2,FILE=LOGBAK,STATUS='NEW')
      OPEN(UNIT=3,FILE=ROKFIL,STATUS='NEW')
      OPEN(UNIT=4,FILE=STRFIL,STATUS='NEW')
      OPEN(UNIT=5,FILE=BITBAK,STATUS='NEW')

      WRITE(1,70)N
C ----- ROCK STRENGTH FILE
      WRITE(3,74)
      WRITE(4,75)
      DO 69 I=1,N
      WRITE(1,71) DEPTH1(I),SHALE(I),0.0,SAND(I),0.0,
     $ LIME(I),0.0,0.0
      WRITE(2,72) DEPTH1(I),0.0,0.0,0.0,0.0,0.0,
     $ 0.0,ROP(I),0.0,WOB(I),RPM(I),0.0
      WRITE(5,73)DEPTH1(I),BTYPE(I),DIA(I),J1(I),J2(I),J3(I),
     $ MW(I),PV(I),GPM(I),COLDIA(I),PIPDIA(I),DRC(I),FRIC(I),YP(I)
C     WRITE(6,74) DEPTH1(I),PP(I)
      WRITE(3,1000) DEPTH1(I),K(I),KI(I),ROCK(I),ROCKI(I),
```

```
    $   BHP(I),BHPI(I)
C ------ INSITU STRESS FILE
      WRITE(4,1001) DEPTH1(I),SHI(I),SH(I),PP(I),BETAP(I),BETAP(I)
69    CONTINUE

      ELSEIF(CODE.EQ.2)THEN

      OPEN(UNIT=3,FILE='DEMO.ROK',STATUS='NEW')
      OPEN(UNIT=4,FILE='DEMO.STR',STATUS='NEW')
      WRITE(3,74)
      WRITE(4,75)
      DO 76 I=1,N
      WRITE(3,1000) DEPTH1(I),K(I),KI(I),ROCK(I),ROCKI(I),
    $   BHP(I),BHPI(I)
C ------ INSITU STRESS FILE
      WRITE(4,1001) DEPTH1(I),SHI(I),SH(I),PP(I),BETAP(I),BETAP(I)
76    CONTINUE

      ENDIF

70    FORMAT(1X,I5)
71    FORMAT(1X,F8.2,2X,7F8.4)
72    FORMAT(1X,F8.2,2X,6F5.1,2X,F7.4,2X,F5.1,2X,F6.3,2X,F7.3,F5.1)
73    FORMAT(1X,14F8.2)
74    FORMAT(T2,'DEPTH',T17,'KP',T28,'KI',T41,'ROCKP',T59,'ROCKI',
    $ T78,'BHPP',T98,'BHPI ')
75    FORMAT(T6,'DEPTH',T17,'SHI',T28,'SHP',T38,'PP',T47,'BETAP',
    $ T57,'BETA4')
1000  FORMAT(1X,F6.1,2X,F10.6,2X,F10.6,2X,F15.3,2X,F15.3,2X,
    $    F15.3,2X,F20.3,2X,F20.3)
1001  FORMAT(2X,4F10.3,2F10.4)

      CLOSE(1)
      CLOSE(2)
      CLOSE(3)
      CLOSE(4)
      CLOSE(5)
      CLOSE(6)
      RETURN
      END□
```

APPENDIX B

COMPUTER PROGRAM: DYNAMIC

```
/* July 16, 1992 -- Gary Bratcher
  This program creates output data files used in the plot routine
  GNUPLOT.  These plots show the variation of

              del h
              --------
              Pbh - Pr

  (The change in a multiple of drill tooth penetration divided by
  the difference between bottom hole pressure and reservoir pressure),
  with drilling parameters.  These drilling parameters are ROP,
  RPM, Bit Diameter, and alpha.  Alpha contains conversion constants,
  permeability, porosity, fluid compressibility.

 */

/*   Some nominal values used are:

    dia     = 8.75 inches
    rpm     = 60 rpm
    k       = 1.0 millidarcy
    phi     = 0.1 vol./vol.
    c       = 10^(-4) (vol*in*in) / (vol*lbf)
    mu      = 1.0 centipoise

    zeta    = 316.  md*hr*lbf
                    ----------
                    cp*ft*in^3

    for these nominal values:
 check
    alpha = 100,000  (md*in*in) / (cp*mu*lbf)

 */

 #include <stdio.h>
 #include <math.h>

 double power(int j)
 {   double n=1;
    int i;
    for(i=0;i<j;i++)
      n=n*10;

    return n;
```

```
}

void vary_k(FILE *ofp, double nD)
{  double k=1.0, phi=0.1, c=0.0001, mu=1.0,a1,a2,a3,a4;
   double term1, term2, zeta=316., alpha;
   double delh, h=0.1, rop;
   int i,j;

   /* plots will all be delh vs. rop, therefore increment other  */
   /* variables and make rop a continuous function */

         a1 = 0.001/(phi*c*mu);
         a2 = 0.01/(phi*c*mu);
         a3 = 0.1/(phi*c*mu);
         a4 = 1.0/(phi*c*mu);
/*  header for stdyalph.dat, with alphasqr values as states */
    fprintf(ofp,"#ROP\t\t1mic\t\t10mic \t\t100mic\t\t1md\n");
    fprintf(ofp,"    \t\t%lf\t\t%lf\t\t%lf\t\t%lf\n",a1,a2,a3,a4);
    fprintf(ofp,"#---\t\t----\t\t-----\t\t-------\t\t---\n\n");

    /* h = 0.1, tooth penetration depth is set at 1/10 in */
    /* three different alphas are used. In all three cases:
           mu = 1.0 cp,
           c  = 0.0001.
           phi= 10 %
    This range of values for k and phi were provided by Ercil Hunt on
    7/17/92, where, for tight gas sands, 10^-3 < k < 1 md */

    for(i=1;i<=1000;i++)
    {
      rop=i;
      fprintf(ofp,"%lf",rop);
      for(j=0;j<4;j++)
      {
          if(j==0) k =0.001;
          if(j==1) k = 0.01;
          if(j==2) k = 0.1;
          if(j==3) k =1.0;
          alpha = k/(phi*c*mu);
          term1=1.0-exp(-1*zeta*rop*h/alpha);
          term2=1.0-exp(-1*zeta*rop*nD/alpha);

          delh=term1/term2;
          fprintf(ofp,"\t%lf",delh);
      }
```

```
    fprintf(ofp,"\n");
  }


}

void main()
{  double nD=0.;
   FILE *al;

   if((al=fopen("nd01.dat","w"))==NULL)
          exit(-2);
   nD = 0.1;
   vary_k(al,nD);
   fclose(al);

   if((al=fopen("nd05.dat","w"))==NULL)
          exit(-2);
   nD = 0.5;
   vary_k(al,nD);
   fclose(al);

   if((al=fopen("nd1.dat","w"))==NULL)
          exit(-2);
   nD = 1.0;
   vary_k(al,nD);
   fclose(al);

   if((al=fopen("nd10.dat","w"))==NULL)
          exit(-2);
   nD = 10.;
   vary_k(al,nD);
   fclose(al);

   if((al=fopen("nd100.dat","w"))==NULL)
          exit(-2);
   nD = 100.;
   vary_k(al,nD);
   fclose(al);
}
```

APPENDIX C

COMPUTER PROGRAM: EXPAND

```
C     11/15/92
C     this version assumes Chorz to be a constant equal to the far
C     field gradient value.

      INTEGER ANSWER
      REAL DIFPRES,DEPTH,CHORZI,CHORZF,PGRAD,MGRAD,LPPG,PERCHG

200   WRITE(*,*) ' ENTER THE DEPTH OF THE BOREHOLE (IN FEET)'
      READ(*,*) DEPTH
      WRITE(*,*) ' '
      WRITE(*,*) ' ENTER THE Chorz COEFFICIENT BEFORE DRILLING'
      WRITE(*,*) ' (HORIZONTAL TO OVERBURDEN RATIO) '
      READ(*,*) CHORZI
      WRITE(*,*) ' '
      WRITE(*,*) ' ENTER THE MUD GRADIANT '
      READ(*,*) MGRAD
      WRITE(*,*) ' ENTER THE PORE PRESSURE GRADIANT '
      READ(*,*) PGRAD

      PERCHG=1.29242460315-0.29748412695*CHORZI
      CHORZF=PERCHG*CHORZI

300   CALL CALPR(DIFPRES,LPPG,DEPTH,CHORZI,CHORZF,MGRAD,PGRAD)


      WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) ' FOR DEPTH = ',DEPTH
      WRITE(*,*) ' AND MUD GRADIANT =',MGRAD
      WRITE(*,*) ' AND PRESSURE GRADIANT =',PGRAD
      WRITE(*,*) ' AND CHORZ INITIAL=',CHORZI
      WRITE(*,*) ' AND CHORZ FINAL  =',CHORZF
      WRITE(*,*) ' '
      WRITE(*,*) ' THE LOCAL PORE PRESSURE GRADIANT =',LPPG
      WRITE(*,*) ' THE DIFFERENTIAL PRESSURE = ',DIFPRES
      WRITE(*,*) ' WITH A PERCENT CHANGE IN CHORZ =',PERCHG
      WRITE(*,*) ' '
      WRITE(*,*) ' '

      WRITE(*,*) ' ENTER A (1) IF YOU WOULD LIKE TO CONTINUE '
      WRITE(*,*) ' OTHERWISE ENTER A (2) '
      READ(*,*) ANSWER

      IF(ANSWER.EQ.1) THEN
        ANSWER=2
```

```
        GOTO 200
      ENDIF

      STOP
      END

      SUBROUTINE
CALPR(DIFPRES,LPPG,DEPTH,CHORZI,CHORZF,MGRAD,PGRAD)
C  10/25/92
C  FORTRAN VERSION OF CCALC.  ITERATES TO AN INITIAL BOTTOMHOLE
C  DIFFERENTIAL PRESSURE.
C

      INTEGER K

      REAL  PP, SIGV0M, SIGH0M, SIGV0P, SIGH0P, DSIG, OLDDSG,
     $      DPP, OLDDPP, CB, CR, CP, PHI, V, E, SMEANM, SMEANP,
     $      MGRAD, PGRAD, DIFPRES, DEPTH, CHORZI,CHORZF,LPPG

      OLDDSG  = 0.0
      OLDDPP  = 0.0
      K       = 0
      E       = 2000000.0
      CR      = 0.000000186
      CW      = 0.0001
      PHI     = 0.15
      V       = 0.25
      CB      = 3.0*(1.0 - 2.0*V)/E
C     CB      = CP*PHI + CR*(1-PHI)


      PP = PGRAD*DEPTH

      SIGV0M = 1.0 * DEPTH
      SIGH0M = CHORZI * SIGV0M

      SMEANM = (SIGV0M+2.0*SIGH0M)/3.0 - PP

C     SIGV0P = 0.052*MW*DEPTH
      SIGV0P = MGRAD*DEPTH
C     MAJOR CHANGE: NOW USING OVERBURDEN PRESSURE INSTEAD OF
MUD COLUMN
C     PRESSURE
      SIGH0P = CHORZF*SIGV0M
```

```
      SMEANP = (SIGV0P+2.0*SIGH0P)/3.0 - PP

      DSIG   = SMEANP - SMEANM

      DPP    = (CB - CR) * DSIG / (PHI*CW + (CB - (1+PHI)*CR))
      PP     = PP + DPP

100   K=K+1

      IF(ABS(OLDDPP-DPP).GT.0.01.AND.ABS(OLDDSG-DSIG).GT.0.01)THEN
         OLDDPP  = DPP
         OLDDSG  = DSIG
         SMEANP  = ( SIGV0P + 2.0*SIGH0P ) / 3.0 - PP
         DSIG    = SMEANP - SMEANM
         DPP     = (CB - CR) * DSIG / (PHI*CW + (CB - (1+PHI)*CR))
         PP      = PP + DPP
         GOTO 100
      ENDIF

      DIFPRES = SIGV0P - PP
      LPPG    = PP / DEPTH

      RETURN
      END□
```

APPENDIX D

TORQUE TEST PROCEDURE

Recommended Procedure to Calibrate Rig Torque in Ft-Lbs Using Rig Pipe Tongs
[Hoberock, Hareland,
and Bratcher, 1992]

The calibration of the rig rotary torque in Ft-Lbs requires a special test utilizing the pipe tong torque on the rig floor working against the rotary drive. The calibration test requires some special instrumentation, as follows:

1. One TOTCO tong torque hydraulic system (TJ-series). This includes a hydraulic piston for measurement of the tension in the long line and a hydraulic gage setup on the rig floor. The hydraulic gage should read line tension in lbs, and the hydraulic pressure range should be 0-5000 psi. (A calibration curve should be produced by TOTCO to verify the linearity of the hydraulic piston and gage, especially in the lower range.)

2. An Exlog transducer that fits the TOTCO hydraulic system and converts the 0-5000 psi hydraulic pressure to an electrical signal that can be recorded by Exlog as a time variable. This transducer should then be connected to the TOTCO system on the rig floor.

3. An Exlog rotary torque sensing system. This system records with time a relative rotary torque. For an electric rig, this system is driven by the current drawn by the electric rotary drive motor. For a mechanical rig, this system is driven by a mechanical\hydraulic idler on the rotary drive chain.

4. Exlog should set up a data collecting scheme where the relative rotary torque and the tong torque in (Ft-Lbs) are collected versus time. (The frequency of data acquisition should be at least once per second, but five per second would be desirable).

The procedure for calibration of the rotary torque in Ft-Lbs is accomplished by having the rotary drive slowly torque up against pipe tongs attached to drill pipe held in the rotary slips and tong line fastened to the rig structure. This calibration should be accomplished after 30 ft has been drilled below an intermediate casing shoe, and also during a trip close to TD. The recommended procedure is:

1. The total length of the drill string should be inside the casing, and a portion of one joint of drill pipe should be above the rotary table. As many drill collars (and heavy weight drill pipe) as possible should be in the casing.

2. The joint of drill pipe above the rotary should be in the slips.

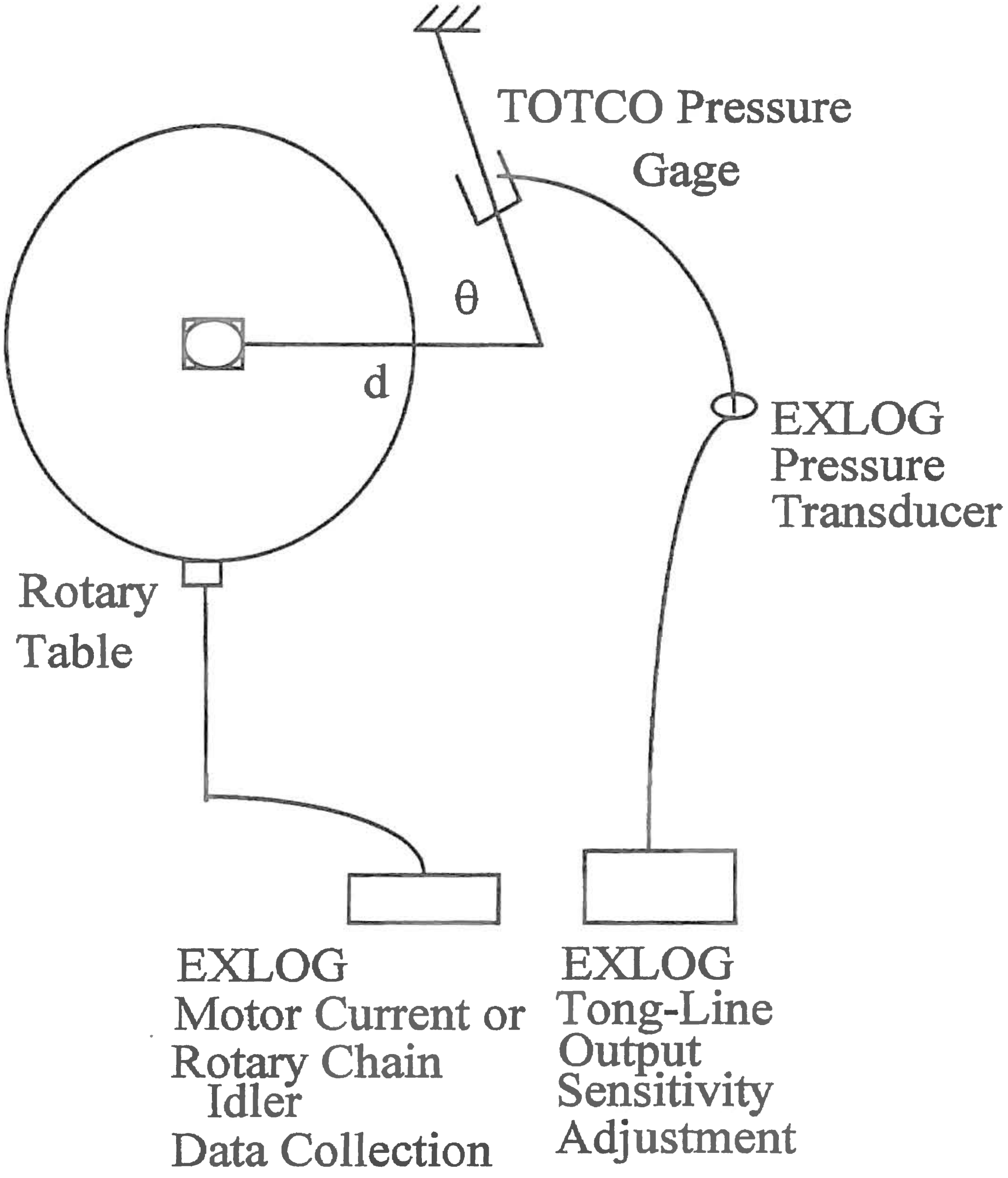
3. The kelly should be connected, but not supporting any weight. The brake should be locked. This is a safety procedure, such that if the slips should jump, the hook and drilling line will catch the drill string.
4. The safety line should be connected to the pipe tongs, but not put in tension.
5. The tong tension line with the TOTCO gage "in-line" is the only line that will work against the rotary.
6. The rotary drive should be slowly activated in low gear, with the tong tension line resisting the turning of the rotary.
7. The Exlog data collection scheme is to collect torque and tong data as power is slowly increased to the rotary drive. The maximum torque applied should NOT exceed maximum drill pipe connection torque.
8. The rotary torque should be applied slowly four times, from zero to different maximum values.
9. Each maximum torque value should be maintained until the reading stabilizes on the TOTCO tong gage on the rig floor.
10. The angle of the tong arm, θ, with respect to its tension line should be measured and recorded in degrees. The length of the tong arm, d from the center of the drill pipe to the tong line should be measured and recorded in inches. (See Figure below).
11. The calibration curve is obtained by correlating the collected maximum rotary torque signal measured by Exlog to the tension in the tong line multiplied by the tong length and the sine of its angle with the tension line, i.e.

$$T = \frac{Fd\sin\theta}{12}$$

where

T = Rotary torque (FT-Lbs)
F = Force in tong tension line from TOTCO gage (Lbs)
d = Length of tong (in)
θ = Angle between tong arm and tong tension line (degrees)

The procedure for the entire test should not take more than 20 minutes.


TOTCO Pressure
Gage
θ
d
EXLOG
Pressure
Transducer
Rotary
Table
EXLOG
Motor Current or
Rotary Chain
Idler
Data Collection
EXLOG
Tong-Line
Output
Sensitivity
Adjustment

APPENDIX E

COMPUTER PROGRAM: E

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <alloc.h>
#define  b1096    1
#define  MODE    10
typedef struct time{
 float t;
 float t0;
 float dt;
 float w;
 float w0;
 float n;
 struct time  *next;
}time;
#include "sasdef.h"
#include "inittw.h"
#define  FALSE    0
#define  TRUE             1
#define  METHOD   "Gauss"


float calc_wc()
{  double num = 0., den = 0.;

  num = (double)(13.33*MU*D*PI*E*A*K1*100.*10000.);
  den = (double)(14400.*L + 13.33*PI*D*MU*E*A*K2*100./10000.);
  if(num>=0.&&den>0.)
     return ((float)sqrt((num/den)));

  return 0;
}

float calc_esmin(float wc)
{  float term1=0.,term2=0.;
  if(wc>0.){
    term1 = (13.33*MU*E*A*K2*100./10000.)/(3600.*D*L);
    term1 = (term1 + 1./ABIT)*wc;
    term2 = (13.33*MU*E*A*K1*100.*10000.)/(3600.*D*L*wc);
    return(term1 + term2);
  }

  return(-1);
}
```

```
void output(time *first_data)
{ time *cur_data=first_data;
 float tcal=0., t=0., w=0., w0=0., rs=0., t0=0.;
 FILE *ofp;

  if ((ofp = fopen(OUT_NAME1, "w")) == NULL)
  {
     fprintf(stderr, "Cannot open output file.\n");
  }

 if(ofp!=NULL){
   while(cur_data!=NULL){
     t = cur_data->t; w = cur_data->w; w0 = cur_data->w0;
     t0 = cur_data->t0; rs = cur_data->n;
     tcal= K1*(1./w - 1./w0)/rs + K2*(w0 - w)/rs + t0;
     fprintf(ofp," %12.2f %12.2f %12.2f %12.2f \n",
     rs*100.,w*10000.,t,tcal);
     cur_data = cur_data->next;
   }
   fclose(ofp);
  }
}


time *mal_dat(int n_el)
{ int i=0;
 time *first_link=NULL, *cur_link=NULL;

 if((first_link =malloc(sizeof(time)))== NULL) return(NULL);
 else{
    cur_link = first_link;
    cur_link->t = 0.; cur_link->t0 = 0.;
    cur_link->w = 0.; cur_link->w0 = 0.;
    cur_link->dt = 0.; cur_link->n = 0.;
    cur_link->next = NULL;
  }

  for(i=1;i<n_el;i++){
    if((cur_link->next = malloc(sizeof(time))) == NULL){
           printf("Not Enough Memory??\n");
           return(NULL);
    }
    else{
           cur_link = cur_link->next;
           cur_link->t = 0.; cur_link->t0 = 0.;
```

```
          cur_link->w = 0.; cur_link->w0 = 0.;
          cur_link->dt = 0.; cur_link->n = 0.;
          cur_link->next = NULL;
    }
  }
  return(first_link);
}

void dump_dat(time *first_data)
{ time *cur_data=first_data, *trash_data=NULL;

   while(cur_data!=NULL){
     trash_data = cur_data;
     cur_data = cur_data->next;
     free(trash_data);
   }
}

void main()
{ int mode = MODE;
  time *first_data=NULL, *cur_data=NULL;
  float tr2=0.,wc=0.,esmin=0.,k1=K1,k2=K2,t=0.,w=0.,rs=0.;
  FILE *ifp,*ofp;

  cur_data = first_data = mal_dat(NDATA);
  if(first_data==NULL) exit(1);


  if ((ifp = fopen(NAME, "r"))== NULL)
  {
    fprintf(stderr, "Cannot open input \
            file.\n");
   }

  /* reads in edited file containing time in seconds and */
  /* wob in lbf. */
   if(ifp!=NULL){
    printf("\nSCANNING FILE\n");
    while(cur_data!=NULL){
         fscanf(ifp," %f %f %f %f ",&t,&tr2,&w,&rs);
         if(rs < 80.){
             cur_data->t = t; cur_data->w = w/10000.;
             cur_data->n = rs/100.; cur_data = cur_data->next;
         }
    }
```

```
    fclose(ifp);
   }
   else exit(1);

  printf("\nINITIALIZING\n");
  init_t0_w0(first_data,mode);

   wc=calc_wc();
   esmin=calc_esmin(wc);

  if ((ofp = fopen(OUT_NAME2, "w"))
    == NULL)
 {
   fprintf(stderr, "Cannot open output file.\n");
 }

 if(ofp!=NULL){
  fprintf(ofp,"\nfile = %s\tDepth = %f\n",OUT_NAME2,L);
  fprintf(ofp,"\nmethod = %s\n",METHOD);
  fprintf(ofp,"\nNUMBER OF DATA POINTS: %d\n\n",NDATA);
  fprintf(ofp," the function is :\n\n\ty = %f (1/W - 1/Wo)/rs + \
  %f (Wo - W)/rs \n",k1,k2);
  fprintf(ofp,"\tWc = %f\t\tEsmin = %f\n",wc,esmin);
    fclose(ofp);
 }

  printf(" the function is :\n\n\ty = %f (1/W - 1/Wo)/rs + %f (Wo - W)/rs\
   \n",k1,k2);
  printf("\tWc = %f\t\tEsmin = %f\n",wc,esmin);

  /* create file */

  printf("\n\nPRINTING OUTPUT FILE\n");
  output(first_data);

  dump_dat(first_data);
 }
```

VITA 2

Gary J. Bratcher

Candidate for the Degree of

Master of Science

Thesis: INVESTIGATION OF ROCK STRENGTH CALCULATIONS USING DRILLING PARAMETERS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Phoenix, Arizona, March 31, 1967, the son of James and Frances Lofton.

Education: Graduated from Shawnee High School, Shawnee, Oklahoma, in May 1985; received Bachelor of Science Degree in Mechanical Engineering from Oklahoma State University in May 1991; completed requirements for the Master of Science Degree at Oklahoma State University in July, 1993.

Professional Experience: Graduate Research Assistant, Mechanical and Aerospace Engineering Department, Oklahoma State University, January 1991-June 1993; Field Engineer, ARKLA Energy Resources, May 1989-August 1989.