Chapter one

Introduction

1-1 Introductory Remarks:

Petroleum reservoirs are found in geologic formations containing porous rock. Porosity φ is the fraction of the rock volume describing the maximum possible fluid volume that can be stored.

1-1-1 Production from Petroleum Reservoirs:

The initial production of hydrocarbons from the underground reservoir is accomplished by the use of natural reservoir energy and referred to as primary production. The oil and gas are displaced to production wells under primary production by (a) fluid expansion, (b) fluid displacement, (c) gravitational drainage. When there is no aquifer and no fluid is injected into the reservoir, the hydrocarbon recovery is brought about mainly by fluid expansion. When there is water influx from the aquifer or when, in lieu of this, water is injected into selected wells, recovery is accomplished by the displacement mechanism, which again may be aided by gravitational drainage. Gas is injected as a displacing fluid to help in the recovery of oil and is also used in gas cycling to recover gas-condensate fluids.

The use of either a natural gas or water injection scheme is called a secondary recovery operation. When a water injection scheme is used as a secondary recovery process, the process is referred as water flooding. The main purpose of either a natural gas or a water injection process is to maintain the reservoir pressure. Hence, the term pressure maintenance program is also used to describe a secondary recovery process.

Other displacement process called tertiary recovery process have been developed for application in situations in which secondary process have become ineffective. However, the same processes have also been considered for reservoir applications when secondary recovery techniques were not used because of low recovery potential. In these later case, the word tertiary is a misnomer [2].

1-1-2 Well Stimulation:

As usual when the well had been drilled , already have cemented , it is becoming ready for production operation for a lot of times , but ; sometimes the production rate decreases as a result of one or more of reasons of production reducing (particle plugging , chemical precipitation ,….etc) , so that we have to solve this problem by using well stimulation , two stimulation treatments are commonly used**:**

- **Acidizing:**

  Matrix stimulation by acidizing is accomplished by injecting chemicals to dissolve and/or disperse materials near the wellbore that impair well production in sandstones or to create new, unimpaired flow channels between the wellbore and a carbonate formation.

- **Hydraulic fracturing:**

  Matrix stimulation by hydraulic fracturing is to split the rock and prop it open with proppants , So stimulation is for:

bypass near or remove well bore damage and return a well to its “natural “productivity / injectivity , To extend a conductive path deep into a formation and thus increase pSroductivity /injectivity beyond the natural level and to produce hydrocarbon from tight formation.

1-2 Problem Statement:

The propagation of the hydraulic fracturing process is complicated; this project is dealing with calculations of hydraulic fracturing width; as a result of simplifying of concept and supporting to simulate programs.

1-3 Project Objectives:

The objectives of this project are:-

1. Development of Matlab programe to calculate hydraulic fracturing width.

2. Obtaining and analyzing results associated with hydraulic fracturing models.

3. Checking and running the programme.

1-4 Methodology Adopted:

The methodology adopted is:-

- Carrying out a comprehensive literature review of the theories of hydraulic fracturing and the methods used in its design.
- The design, implementation and application of hydraulic fracturing computer programme by using MatLab.
- Verifying the obtained results by comparison with known results

1-5 Outlines of project:

Chapter One presents general introductory remarks, problem statement, objectives and methodology

Chapter Two includes introduction, fracturing, the fracturing process, reasons of fracture models, the Khristianovic and Zheltov (1955) and Geertsma and de Klerk(1969) (KGD) model, Perkins and Kern (1961) and Nordgren (1972) (PKN) model .

Chapter three includes the programmre flow chart, brief description of the programme, implementation of programme, general description of case study.

Chapter four includes the results obtained from the programme.

Chapter five includes comments, conclusions and recommendations.

Chapter Two

Literature Review

2-1 Introduction:

Hydraulic fracturing is an important and prevalent process it has been used in petroleum engineering for more than five decades to create deep-penetrating fractures in hydrocarbon reservoirs stimulating the production of oil and gas.

Hydraulic fracturing in solid, brittle materials (e.g., rock) has been studied extensively. Hence, a relatively good understanding of these brittle hydraulic fractures exists *the term "hydraulic fracture "can be defined* as “the condition leading to the creation and propagation of a thin physical separation in a soil or rock mass due to high fluid pressure [9].

2-2 Fracturing:

If fluid is pumped into a well faster than the fluid can escape into the formation, inevitably pressure rises and at some point something breaks. Because rock is generally weaker than steel what breaks is usually the formation resulting in the well bore splitting along its axis as a result of tensile hoop stresses generated by the internal pressure. The simple idea of the well bore splitting like a pipe becomes more complex for cased and perforated wells and non vertical wells. However, in general the well bore breaks. The rock fracture owing to the action of the hydraulic fluid pressure and a hydraulic fracture is created [5].

2-3 Reasons for Fracture:

Hydraulic fracture operations may be performed on a well for one (or more) of the following three reasons:

- To by- pass near well bore damage and return a well to its natural productivity.
- To extend a conductive path deep into a formation and thus increase productivity beyond the natural level.
- To alter fluid flow and produce from tight formation.

2-4 Hydraulic fracturing process:

In order to carry out hydraulic fracturing operations, a fluid must be pumped into the well's production casing at high pressure.

It is necessary that production casing has been installed and cemented and that it is capable of withstanding the pressure that it will be subjected to during hydraulic fracture operations. In some cases, the production casing will never be exposed to high pressure except during hydraulic fracturing. In these cases, a high pressure "fracturing" may be used to pump the fluids into the well to isolate the production casing from the high treatment pressure.

The well operator or the operator's designated representative should be on site throughout the hydraulic fracturing process. Prior to beginning the hydraulic fracture treatment, all equipment should be tested to make sure it is in good operating condition. All high-pressure lines leading from the pump trucks to the wellhead should be pressure tested to the maximum treating pressure. Any leaks must be eliminated prior to initiation of the hydraulic fracture treatment. When these conditions are met, the well is ready for the hydraulic fracturing process. In the field, the process is called the "treatment" or the "job." The process is carried out in predetermined stages that can be altered depending on the site-specific conditions or if necessary during the treatment. In general, these stages can be described as follows:

> **Pad—**the pad is the first stage of the job. The fracture is initiated in the targeted formation during the initial pumping of the pad. From this point forward, the fracture is propagated into the formation. Typically, no proppant is pumped during the pad; however, in some cases, very small amounts of sand may be added in short bursts in order to abrade or fully open the perforations.

Another purpose of the pad is to provide enough fluid volume within the fracture to account for fluid leak-off into the targeted formations that could occur throughout the treatment.

> **Proppant Stages—**after the pad is pumped, the next stages will contain varying concentrations of proppant.

The most common proppant is ordinary sand that has been sieved to a particular size.

Other specialized proppants include sintered bauxite, which has an extremely high crushing strength, and ceramic proppant, which is an intermediate strength proppant.

> **Displacement—**the purpose of the displacement is to flush the previous sand laden stage to a depth just above the perforations. This is done so that the pipe is not left full of sand, and so that most of the proppant pumped will end up in the fractures created in the targeted formation. Sometimes called the flush, the displacement stage is where the last fluid is pumped into the well. Sometimes this fluid is plain water with no additives, or it may be

The same fluid that has been pumped into the well up to that point in time[23].

Pressure behavior during a fracturing treatment is illustrated by Figure 2-3. The fluid injection rate is constant except that at some time injection is stopped to obtain the instantaneous shut in pressure .The bottom hole pressure is shown versus time from the initial injection of fluid until the treatment has been completed .The critical portions of the pressure history shown in Figure2-7 are :

- Breakdown pressure: the pressure required to break down the formation and initiate fracture.
- Propagation pressure: the pressure required to continually enlarge the fracture.
- Pumping stops: the pressure drops suddenly to a lower value but continues to decrease slowly to the reservoir pressure due to fluid leaking from the fracture.
- Instantaneous shut in pressure: the pressure that is required to just hold the fracture open.

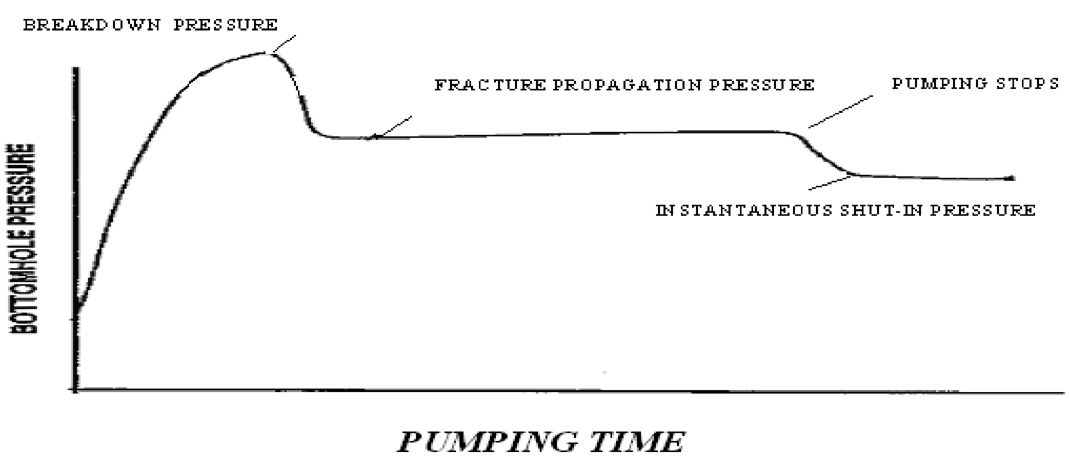


Figure (2-1): Idealized Pressure Behavior during Fracturing [8]

In general hydraulic fracturing treatments are used to increase the productivity index of a producing well or the injectivity index of an injection well. The productivity index defines the rate at which oil or gas can be produced at a given pressure differential between the reservoir and the well bore. The injectivity index refers to the rate at which fluid can be injected into a well at a given pressure differential [8].

2-5 Hydraulic fracturing equipment and materials:

The hydraulic fracturing process requires an array of specialized equipment and materials. This section will describe the materials and equipment that are necessary to carry out typical hydraulic fracture operations in vertical and horizontal wells.

The equipment required to carry out a hydraulic fracturing treatment includes fluid storage tanks, proppant transport equipment, blending equipment, pumping equipment, and all ancillary equipment such as hoses, piping, valves, and manifolds. Hydraulic fracturing service companies also provide specialized monitoring and control equipment that is necessary in order to carry out a successful treatment [23].

2-6 Fracture shapes:

2-6-1 Horizontal Fracture:

Hydraulic fractures are formed in the direction perpendicular to the least stress. In Figure (2-1), an imaginary cube of rock is shown as having confining stress exerted on it in three dimensions. Each pair of opposing stresses must be equal in order for the cube to remain stationary in space. The relative size of the arrows represents the magnitude of the confining stress. In Figure (2-2), the least stress is in the vertical direction. This direction is known as the direction of overburden, referring to the weight of the earth that lies above. The Earth's overburden pressure is horizontal fractures will occur at depths less than 2000 ft.

In this example, when pressure is applied to the center of this block, the formation will crack or fracture in the horizontal plane as shown, because it will be easier to part the rock in this direction than any other direction.

In general, these fractures are parallel to the bedding plane of the formation.

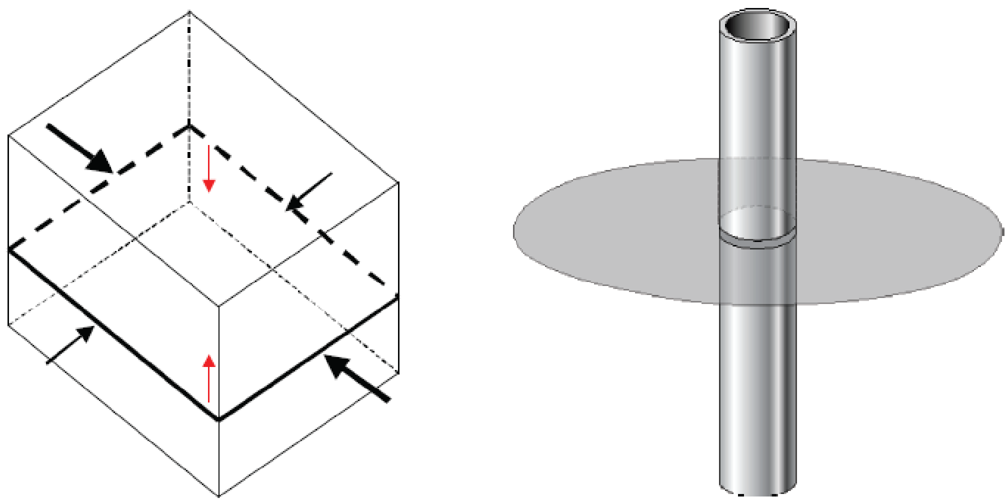


Figure (2-2) Horizontal Fracture [23]

2-6-2 Vertical fracture:

As depth increases, overburden stress in the vertical direction increases by approximately 1 psi/ft. As the stress in the vertical direction becomes greater with depth, the overburden stress (stress in the vertical direction) becomes the greatest stress. This situation generally occurs at depths greater than 2000 ft. This is represented in Figure 2-2 by the magnitude of the arrows, where the least stress is represented by the small horizontal arrows, and the induced fracture will be perpendicular to this stress, or in the vertical orientation.

Since hydraulically induced fractures are formed in the direction perpendicular to the least stress, as depicted in Figure 2-2, the resulting fracture would be oriented in the vertical direction.

The extent that the created fracture will propagate in the vertical direction toward a USDW is controlled by the upper confining zone or formation.

This zone will stop the vertical growth of a fracture because it either possesses sufficient strength or elasticity to contain the pressure of the injected fluids**.**

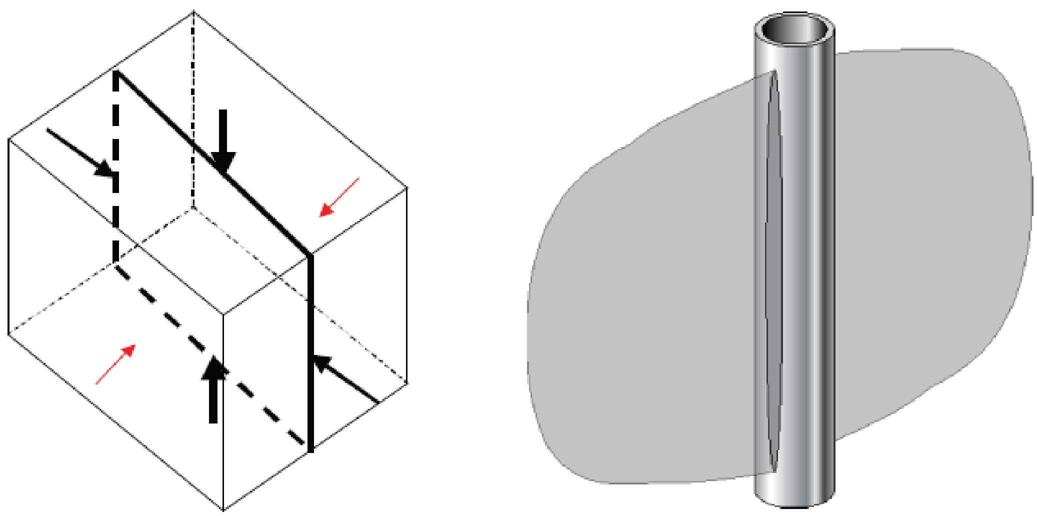


Figure (2-3) Vertical Fracture[23]

2-7 Hydraulic Fracturing Models:

Following the fracture initiation, additional fluid injection would result in fracture propagation. The geometry of the created fracture can be approximated by models that take into account the mechanical properties of the rock, the properties of the fracturing fluid, the conditions with which the fluid is injected (rate, pressure), and the stresses distribution in the pours medium.

Three general families of models are available: two dimensional (2-D), pseudo-three-dimensional (p-3-D) and fully three-dimensional (3-D) models. The latter allow full three-dimensional fracture propagation with full two-dimensional fluid flow.

Two-dimensional models are closed-form analytical approximations assuming constant and known fracture height. For petroleum engineering applications, two mutually exclusive models have been used. For a fracture length much larger than the fracture height the Perkins and Kern (1961) and Nordgren (1972)[14] or PKN model is an appropriate approximation. For a fracture length less than the fracture height the appropriate model has been presented by Khristianovic and Zheltov (1955) [17] and Geertsma and de Klerk(1969)[13]. This is frequently known as the KGD model. A limiting case, where the a fracture length equal double the fracture height is the radial or "penny-shape" model. The fracture height used here is the dynamic value, that is the fracture height at the time that the fracture length **.**

2-7-1 The Khristianovic-Geertsma-de Klerk(KGD) Model:

The Khristianovic and Zheltov (1955) and Geertsma and de Klerk(1969)[13] (KGD) fracture model is shown in Figure 2-3.In addition to the constant height assumption, the two other assumptions are:-

1) The fracture is at a plane strain condition in the horizontal plane.

2)The fracture tip is a cusp-shaped tip as proposed by Barenblatt. This assumption of a cusp-shaped tip removes the stress singularity at the fracture tip which would otherwise be predicted by the elasticity analysis**.**

2) Fixed fracture height, hf.

3) Rock stiffness is taken into account in the horizontal plane only. 2D plane strain deformation in the horizontal plane.

4) Thus fracture width does not depend on fracture height and is constant in the vertical direction.

5) The fluid pressure gradient is with respect to a narrow, rectangular slit of variable width .

6) The shape of the fracture in the horizontal plane is elliptic with maximum width at the wellbore narrow, rectangular slit of variable width.

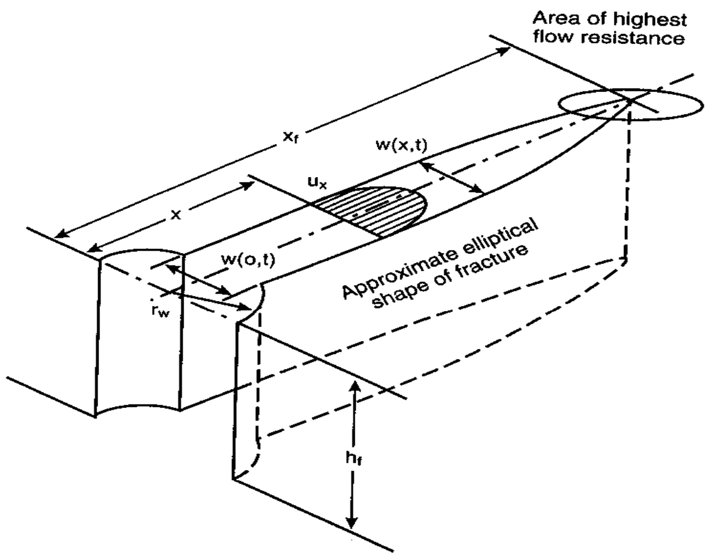


Figure (2-3). The KGD Model Geometry[13]

Following Geertsma –de Klerk[13], the fracture is approximated as a channel of opening width w. The pressure distribution for the flow of a viscous fluid inside the fracture can be written as:

$$p_w - p = \frac{12\mu QL}{h} \int_{f_{Lw}}^{f_L} \frac{df_L}{w^3} \text{ ------------------- (2-1)}$$

Where:

$f_L = x/L$,

$f_{Lw} = r_w/L$

h = fracture height,

L= total length of the fracture,

p = local fluid pressure ,

p_w = fluid pressure at well bore,

Q = fluid injection rate,

w = local fracture width,

μ = frac-fluid viscosity,

r_w = well bore radius.

The above equation has two unknowns; p and w. England and Greens solution for a plane fracture in an infinite elastic medium provides another relationship between p and w as:

$$w = \frac{4(1-\nu)L}{\pi G}\left[\int_{f_L}^{1} \frac{f_2 df_2}{\sqrt{f_2^2 - f_L^2}} \int_{0}^{f_2} \frac{p(f_1)df_1}{\sqrt{f_2^2 - f_1^2}} - \frac{\pi}{2}\sigma_{\min}\sqrt{1-f_L^2}\right] \text{-------------(2-2)}$$

where:

G and ν = shear modulus and Poisson's ratio of rock, respectively,

f_1 And f_2 = fraction of fracture extent,

$\sigma_{\min}$ =the minimum in-situ stress.

The time history of fracture width w(t) and fluid pressure p(t) can be obtained by solving eqn 2-1 and eqn 2-2 with proper boundary conditions, such as:

$$\left(\frac{dw}{df_L}\right)_{f_L=1} = 0 \text{ ----------------------------------(2-3)}$$

By assuming that the dry zone in front of fracture tip is small and that the shape of the wet portion in fracture can be approximated by an ellipse, the following approximate solutions are obtained by Geertsma and de Klerk[13]:

Fracture length:

$$L = 0.48[\frac{8GQ^3}{(1-\nu)\mu}]^{(\frac{1}{6})} t^{(\frac{2}{3})} \text{ --------------------------------------(2-4)}$$

Maximum fracture opening width:

$$w_o = 1.32[\frac{8(1-\nu)Q^3\mu}{G}]^{(\frac{1}{6})} t^{(\frac{1}{3})} \text{ ---------------------------------(2-5)}$$

Well bore pressure:

$$p_w = \sigma_{\min} + 0.96[\frac{2G^3Q\mu}{(1-\nu)^3 L^2}]^{(\frac{1}{4})} \text{ --------------------------------(2-6)}$$

2-7-2 The Perkins –Kern-Nordgren (PKN) Model:

Figure 2-4 is a sketch of a Perkins and Kern (1961) and Nordgren (1972) (PKN) fracture[14].

In addition to assuming a constant fracture height, the other two assumptions are:

1) The fracture is at a state of plane strain in the vertical plane and the vertical fracture cross –section is elliptical.

2) The fracture toughness has no effect on the fracture geometry.

3) The fracture height, hf, is fixed and independent of fracture length.

4) The fracture fluid pressure is constant in the vertical cross sections perpendicular to the direction of propagation.

5) Reservoir rock stiffness, its resistance to deformation prevails in the vertical plane; i.e., 2D plane-strain deformation in the vertical plane

Each plane obtains an elliptic shape with maximum width in the center.

6) The fluid pressure gradient in the x direction can be written in terms of a narrow, elliptical flow channel.

7) The fluid pressure in the fracture falls off at the tip, such that .

8) Flow rate is a function of the growth rate of the fracture width**.**

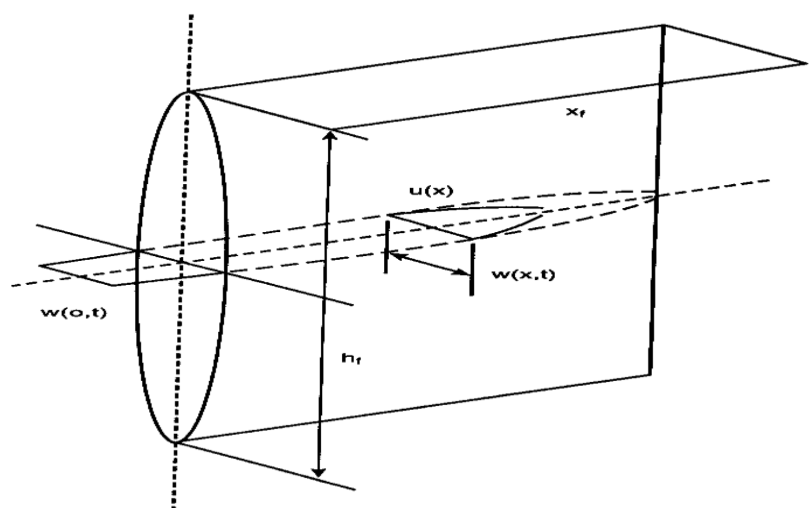


Figure (2-4) the PKN Model Geometry[14]

Following Nordgren[14], the continuity equation for flow of an incompressible fluid inside the fracture can be written as:

$$\frac{\partial q}{\partial x} + q_l + \frac{\partial A}{\partial t} = 0 \text{ --------------------------------(2-7)}$$

where:

q (x,t)=volume rate of flow through a cross-section of the fracture,

q_l (x,t)=volume rate of fluid leak- off per unit fracture length,

A(x,t) = cross-sectional area of the fracture.

The elliptical fracture opening width w is directly related to the net pressure p by the equation:

$$w = \frac{(1-\nu)}{G}(h^2 - 4z^2)^{(\frac{1}{2})} p \text{ -------------------------(2-8)}$$

Knowing the fracture geometry, the fracture cross-sectional area can be written as:

$$A = \int_{\frac{-h}{2}}^{\frac{h}{2}} w dz = \frac{\pi}{4} W h \text{ --------------------------(2-9)}$$

Where, W=is the maximum fracture opening width.

The volume rate of fluid flow q is related to the pressure gradient by the solution for laminar flow of Newtonian fluid in an elliptical tube to give:

$$q = -\frac{\pi W^3 h}{64\mu}\frac{\partial p}{\partial x} \text{ -----------------------------------(2-10)}$$

The fluid leak off rate is expressed as:

$$q_l = \frac{2c_l h}{\sqrt{t-\tau(x)}}$$ ------------------------------------(2-11)

where:

c_l = fluid loss coefficient,

$\tau(x)$ = time at which fluid leak-off begins at position x.

Substitution of eqns 2-9,2-10and 3-11 into eqn 2-7 gives the governing equation for the propagation of a hydraulically induced fracture as:-

$$\frac{G}{64(1-\nu)\mu h}\frac{\partial^2 W^4}{\partial x^2} = \frac{8c_l}{\pi\sqrt{t-\tau(x)}} + \frac{\partial W}{\partial t}$$ -------------(2-12) The initial condition for the eqn 3-12 is:

$$W(x,0) = 0$$ --(2-13)

and the boundary conditions are:

$W(x,t) = 0$ at x $\supseteq$ L(t) fracture length to be determined as a part of the solution.

$$[\frac{\partial W^4}{\partial x}]_{x=0} = -\frac{256(1-\nu)\mu}{\pi G}Q$$ ----------(2-14)

The above equations were solved numerically by Nordgren[14]. It is interesting to note that the well bore pressure predicted by the PKN model, in contrast to the KGD model, increases as the fracture length increases. In the extreme cases of small and large fluid leak-off, an analytical solution can be derived from eq 2-12 as follows:

a- For a large fluid leak-off :

Fracture length:

$$L = \frac{Q}{\pi c_l h} t^{\frac{1}{2}} \text{ ----------------------------------- (2-15)}$$

Fracture opening width:

$$w_o = 4[\frac{2(1-\nu)\mu Q^2}{\pi^3 G c_l h}]^{\frac{1}{4}} t^{\frac{1}{8}} \text{ ------------------------(2-16)}$$

Well bore opening width:

$$p_w = 4[\frac{2G^3 \mu Q^2}{\pi^3 (1-\nu)^3 c_l h^5}]^{\frac{1}{4}} t^{\frac{1}{8}} \text{ --------------------------(2-17)}$$

b- For no fluid leak-off:

Fracture length:

$$L = 0.68[\frac{GQ^3}{(1-\nu)\mu h^4}]^{\frac{1}{5}} t^{\frac{4}{5}} \text{ ------(2-18)}$$

Fracture opening width:

$$w_o = 2.5[\frac{(1-\nu)\mu Q^2}{Gh}]^{\frac{1}{5}} t^{\frac{1}{5}} \text{ ---------------------------(2-19)}$$

Well bore pressure (net):

$$p_w = 2.5[\frac{G^4 \mu Q^2}{(1-\nu)^4 \ h^6}]^{\frac{1}{5}} t^{\frac{1}{5}}$$ --------------------------------(2-20)

All the above PKN and KGD equations have been used after substituting the associated factors which represent a flow chart mentioned in chapter four. The substituted factors include the following:

1- Flow rate
2- Leak-off time
3- Poisson's ratio
4- Young modulus
5- Leak-off coefficient
6- Time
7- Viscosity
8- Density

Chapter Three

Matlab Programme Development and Implementation

3-1 Introduction:

As stated in chapter two, the programme was developed using MatLab.

3-2 Brief Description of programme:

Figure 3-1shows the flow chart of the programme. In the following sections a brief description of the programme is presented.

Referring to the flow chart (Figure 3-1):-

Part A:

In this part, the input data is read according to rock properties (Young's models, fluid viscosity, Poison's ratio, time at which fluid leak off begins, leak off coefficient), pumping schedule (number of stage, injection start time, injection rate, fluid density)

Part B:

The calculation of shear modulus and volume rate of fluid leak off.

Part C:

The quantity of volume rate of fluid leak off (q_e) is evaluated. If it is not equal to zero that indicate for fracture fluid leaking, so go to part D, otherwise go to part E.

Part D:

Calculation of hydraulic fracture width and length for a large fluid fluid leak-off based on the PKN model

In this part, length/height ratio of fracture determined

Part F:

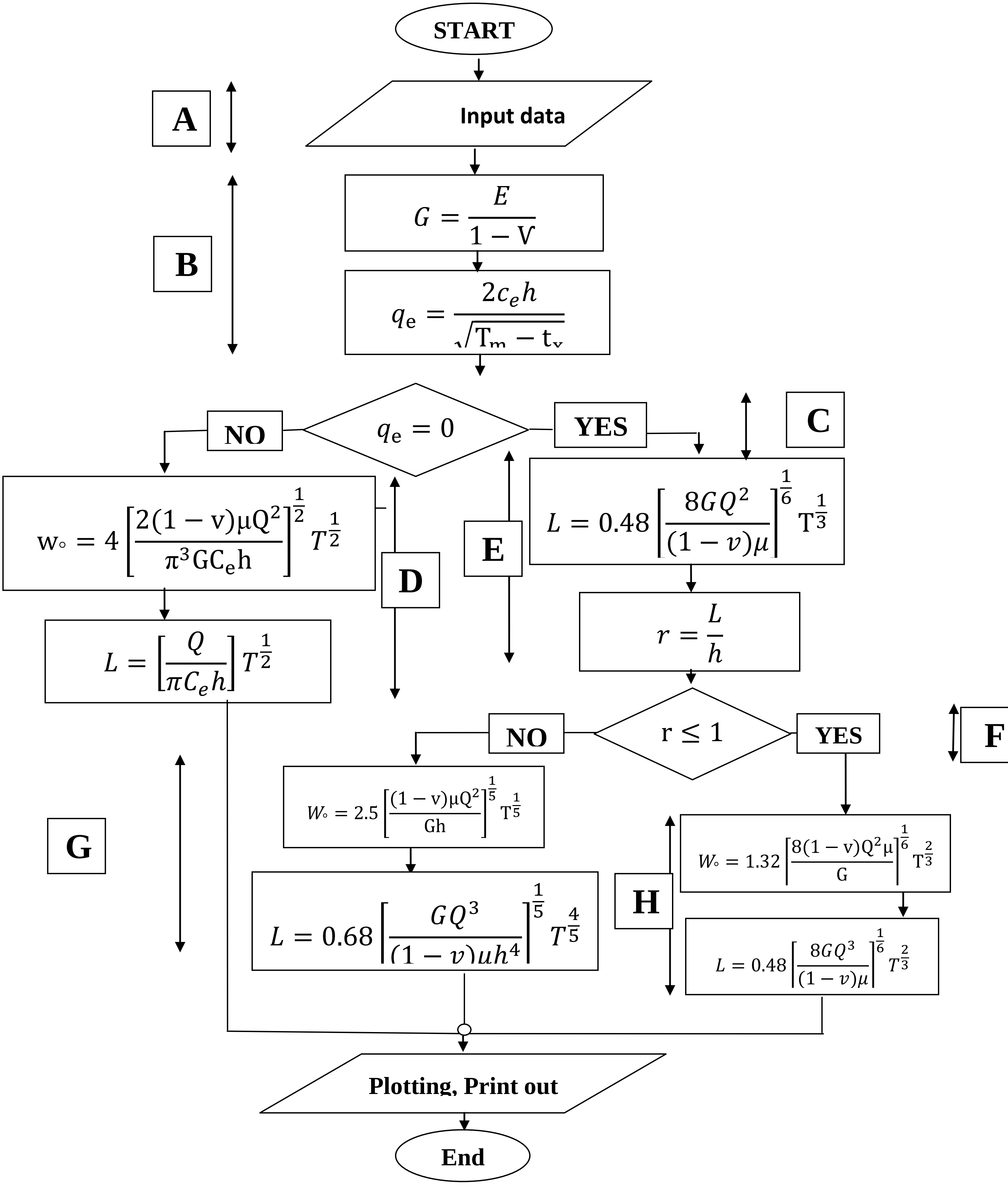
The length/height ratio evaluated. If it is greater than unity go to part G or go to part H.

Part G:

Calculation of hydraulic fracture width and length for no fluid leak-off based on the PKN model.

Part H:

Calculation of hydraulic fracture width and length for no fluid leak-off based on the KGD model.


START
Input data
A
B
$G = \frac{E}{1 - V}$
$q_e = \frac{2c_e h}{\sqrt{T_m - t_x}}$
$q_e = 0$
NO
YES
C
D
E
$w_\circ = 4\left[\frac{2(1-v)\mu Q^2}{\pi^3 G C_e h}\right]^{\frac{1}{2}} T^{\frac{1}{2}}$
$L = \left[\frac{Q}{\pi C_e h}\right] T^{\frac{1}{2}}$
$L = 0.48\left[\frac{8GQ^2}{(1-v)\mu}\right]^{\frac{1}{6}} T^{\frac{1}{3}}$
$r = \frac{L}{h}$
$r \leq 1$
NO
YES
F
G
H
$W_\circ = 2.5\left[\frac{(1-v)\mu Q^2}{Gh}\right]^{\frac{1}{5}} T^{\frac{1}{5}}$
$L = 0.68\left[\frac{GQ^3}{(1-v)\mu h^4}\right]^{\frac{1}{5}} T^{\frac{4}{5}}$
$W_\circ = 1.32\left[\frac{8(1-v)Q^2\mu}{G}\right]^{\frac{1}{6}} T^{\frac{2}{3}}$
$L = 0.48\left[\frac{8GQ^3}{(1-v)\mu}\right]^{\frac{1}{6}} T^{\frac{2}{3}}$
Plotting, Print out
End

3-3 Implementation of Programme:

Based on the flow chart Figure (3-1) and using MatLab the programme was implemented on the computer as shown in the Figure (3-2) and appendix A.

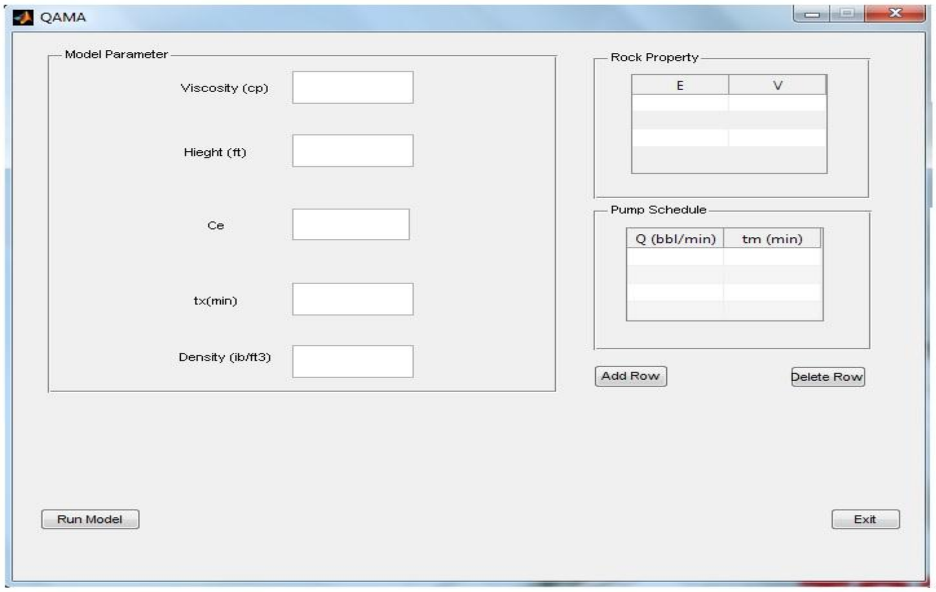


Figure (3-2) The Programme Main Screen

The programme was then applied to obtain a solution as presented in the following sections.

3-4 General Description of Case Study:

Two-sample calculations are carried out. The first case occurs when the in-situ stress in the perforated zone has a contrast of 50 Psi and a gradient of 0.8 Psi/ft with pump schedule change effect on the width, The second case occurs when the in-situ stress in the perforated zone has a contrast of 50 Psi and a gradient of 0.8

Psi/ft with rock properties change effect on the width and The rock property, slurry property, and pumping schedule are listed in table (3-1).

Table (3-1): Input Data to the Programme

Rock Properties:				
Young's Models		10*105 Psi		
Fluid viscosity		150 cp		
Poisson's ratio		0.3		
Fracture toughness		1000 lb/in1.5		
Leak off coefficient		0.0005 ft/min0.5		
Pump Schedule:				
Stage	Inj. Start	Inj. Rate	Fluid density	prop density
No	Time (min)	bbl/min	lb/ft3	lb/ft3
1	0.0	20.0	62.0	0
2	10.0	20.0	62.0	150
3	20.0	20.0	62.0	150
4	26.0	20.0	62.0	150

Chapter Four

The Results

4-1 Results:

Solutions were then obtained for time increments from 0 to 30 mints for pressure of 50 Psi and a gradient of 0.8 Psi/ft

Samples of the results obtained are shown in Table (4-1) and Figure (4-1) compared of with result from reference.

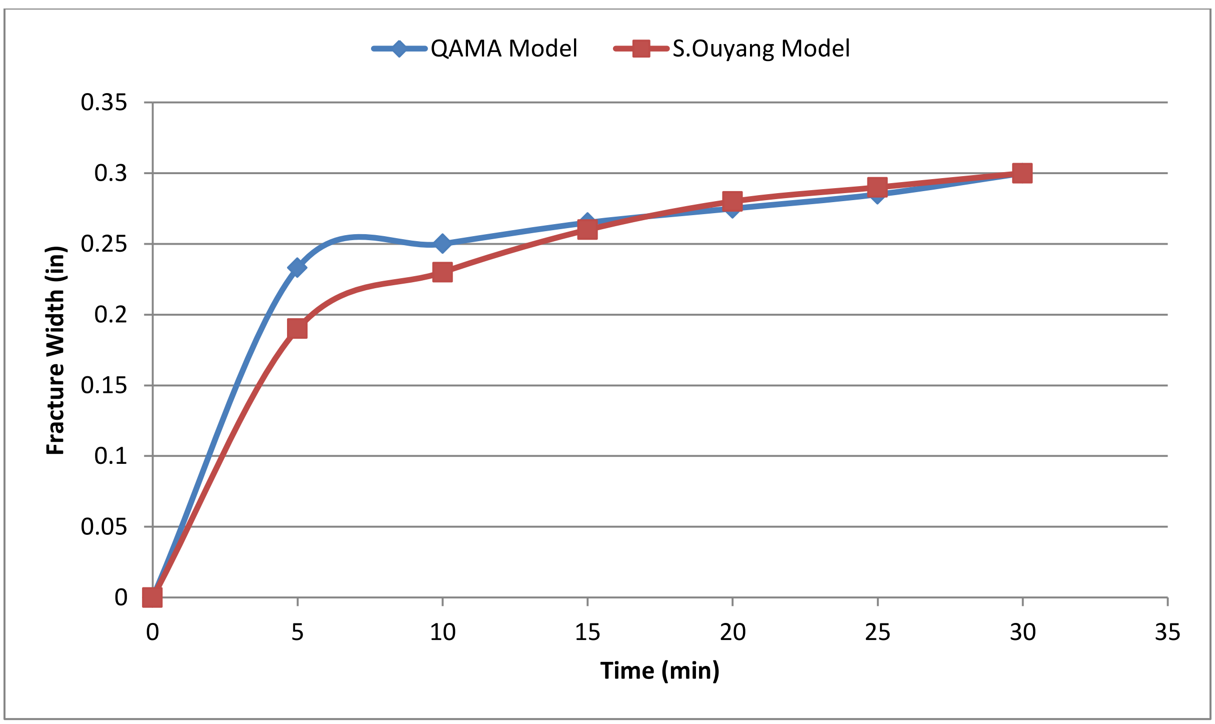
Table (4-2) and Figure (4-2) shows the fracture width for case 1 in which the in situ stress distribution has a contrast of 50 psi and a gradient of 0.8 psi/ft along the perforated zone and pump flow rates of 40, 30 and 20 bbl/min pump schedule effect.

Table (4-3a) and Figure (4-3a) shows the fracture width for case 2 in which the in situ stress distribution has a contrast of 50 psi and a gradient of 0.8 psi/ft along the perforated zone and poison's ratios of 0.3, 0.25 and 0.20 (the rock properties effect).

Table (4-3b) and Figure (4-3b) shows the fracture width for case 2 in which the in situ stress distribution has a contrast of 50 psi and a gradient of 0.8 psi/ft along the perforated zone and Young's Models of $1*10^6$, $1.5*10^6$ and $2*10^6$ (the rock properties change).

Table (4-1) the results of hydraulic fracture width with reference model.

Time (min)	Fracture Width(in) using QAMA model	Fracture width(in) using S.Ouyang model
0	0	0
5	0.233	0.19
10	0.234	0.23
15	0.265	0.26
20	0.275	0.28
25	0.285	0.28
30	0.30	0.30



Figure(4-1)Fracture Width vs Time compared with the references (pressure 50 psi)

Table (4-2) results of case1 for pump flow rate of 40, 30 and 20 bbl/min

Time	Fracture Width(in) (Q=40 bbl/min)	Fracture Width(in) (Q=30 bbl/min)	Fracture Width(in) (Q=20 bbl/min)
0	0	0	0
5	0.33	0.28	0.23
10	0.36	0.31	0.25
15	0.38	0.33	0.27
20	0.39	0.34	0.28
25	0.40	0.35	0.28
30	0.41	0.36	0.29

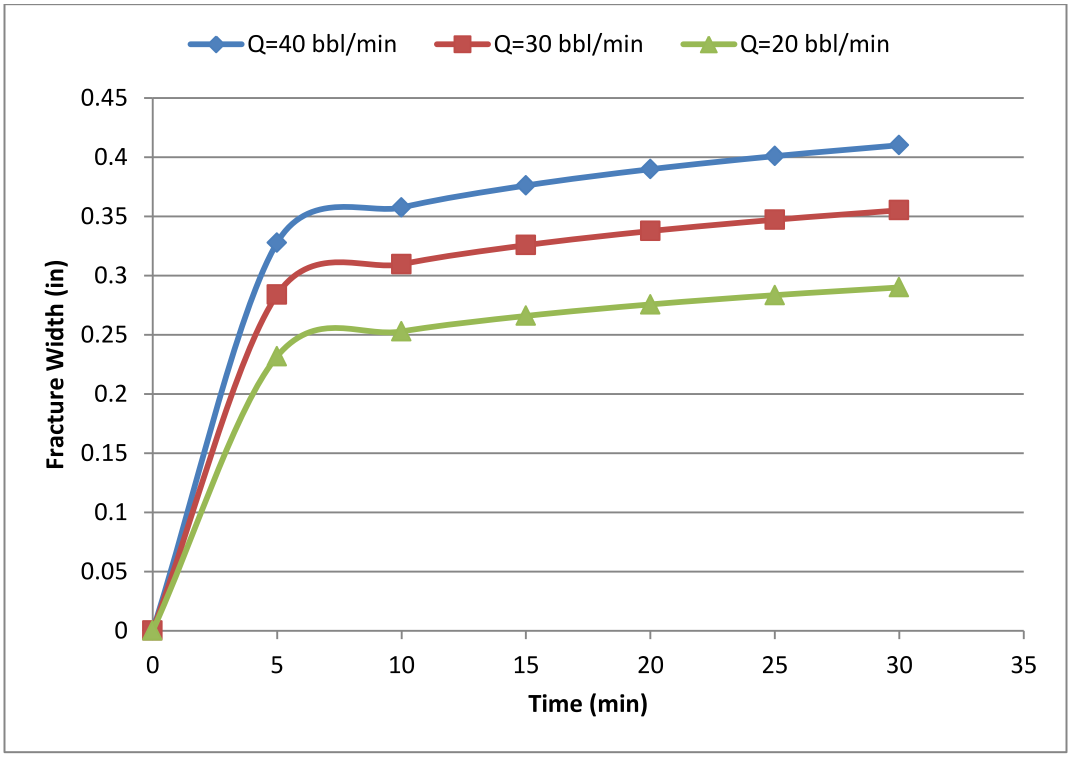


Figure (4-2) Time vs. Fracture Width with flow rate change.

Figure below show that the hydraulic fracture shape is in a vertical plane and the model used to calculate width is PKN model (leak off).

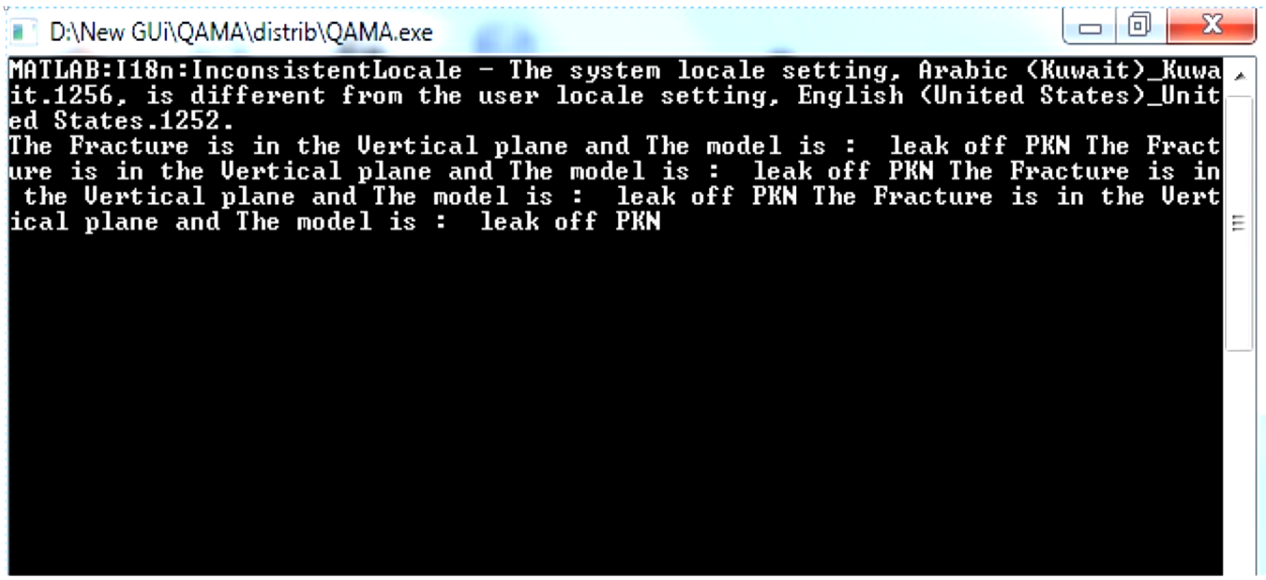


Figure (4-3) fracture shape and the model being used.

Table (4-3a) results of hydraulic fracture width for case2 v= 0.30, 0.25, and 0.20:

Time(min)	Fracture Width(in) (v=0.30)	Fracture Width(in) (v=0.25)	Fracture Width(in) (v=0.20)
0	0	0	0
5	0.23	0.24	0.24
10	0.25	0.26	0.26
15	0.27	0.27	0.28
20	0.28	0.28	0.29
25	0.28	0.29	0.29
30	0.29	0.30	0.30

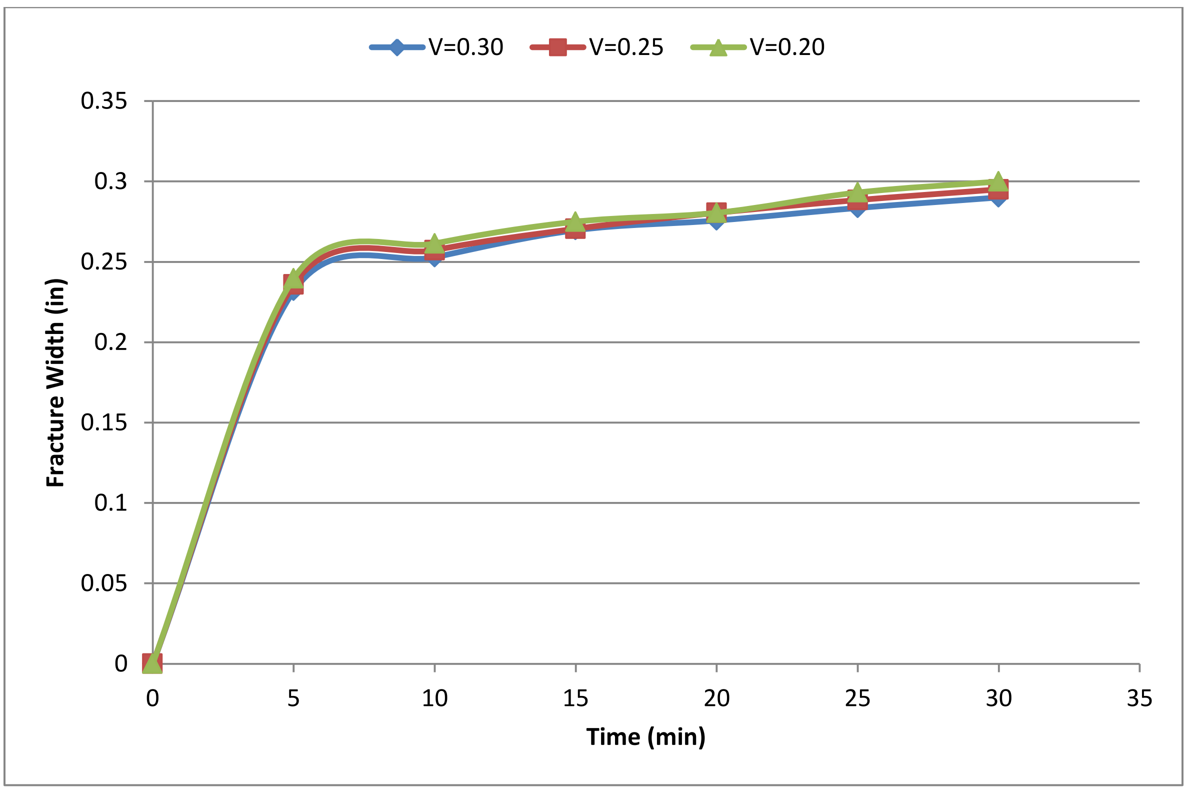


Figure (4-4a) Time vs. Fracture Width for case2 (Poisson's ratio change).

Figure below show that the hydraulic fracture shape is in a vertical plane and the model used to calculate width is PKN model (leak off).

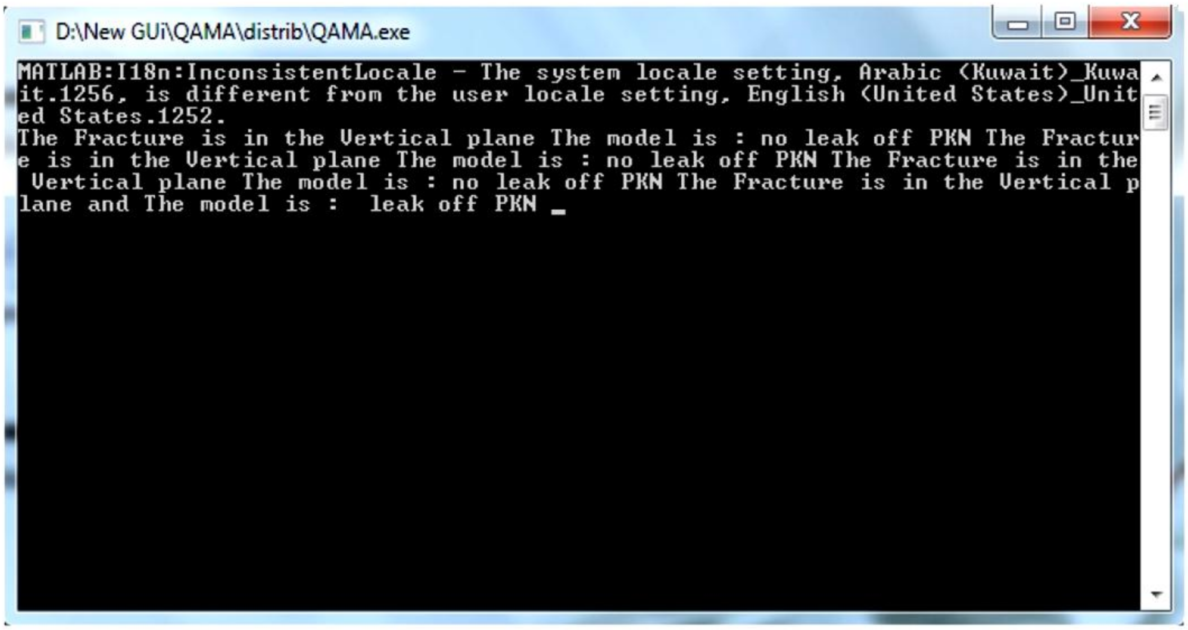


Figure (4-5) fracture shape and the model being used

Table(4-3b) results of Fracturing Width for case2 E=1*10^6, 1.5*$10^6$and 2*$10^6 psi$

Time(min)	Fracture Width(in) (E=1*10^6psi)	Fracture Width(in) (E=1.5*10^6psi)	Fracture Width(in) (E=2*10^6psi)
0	0	0	0
5	0.72	0.66	0.63
10	0.83	0.76	0.72
15	0.89	0.83	0.78
20	0.95	0.87	0.83
25	0.99	0.91	0.86
30	1.03	0.95	0.89

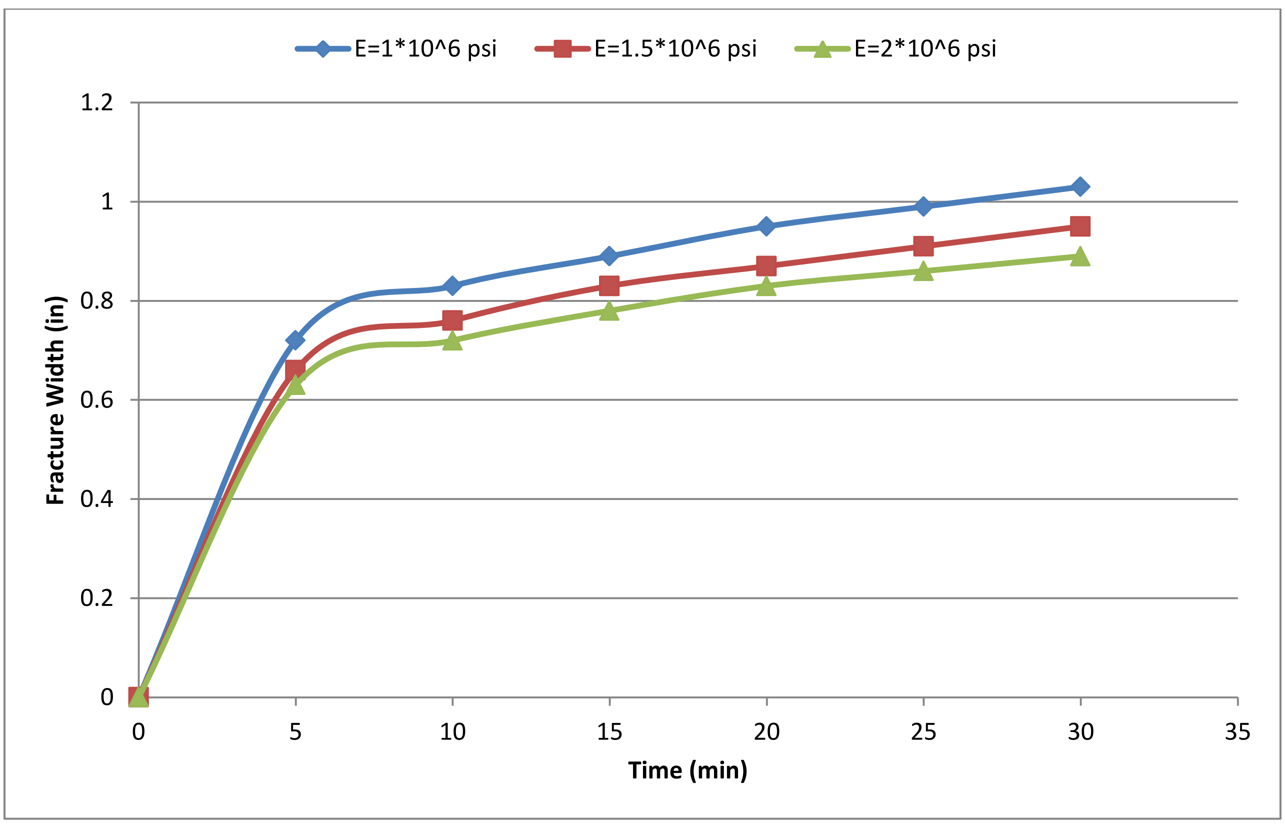


Figure (4-4b) Times vs. Fracture Width for case2 (Young’s Models change).

Figure below show that the hydraulic fracture shape is in a vertical plane and the model used to calculate width is PKN model (no leak off).

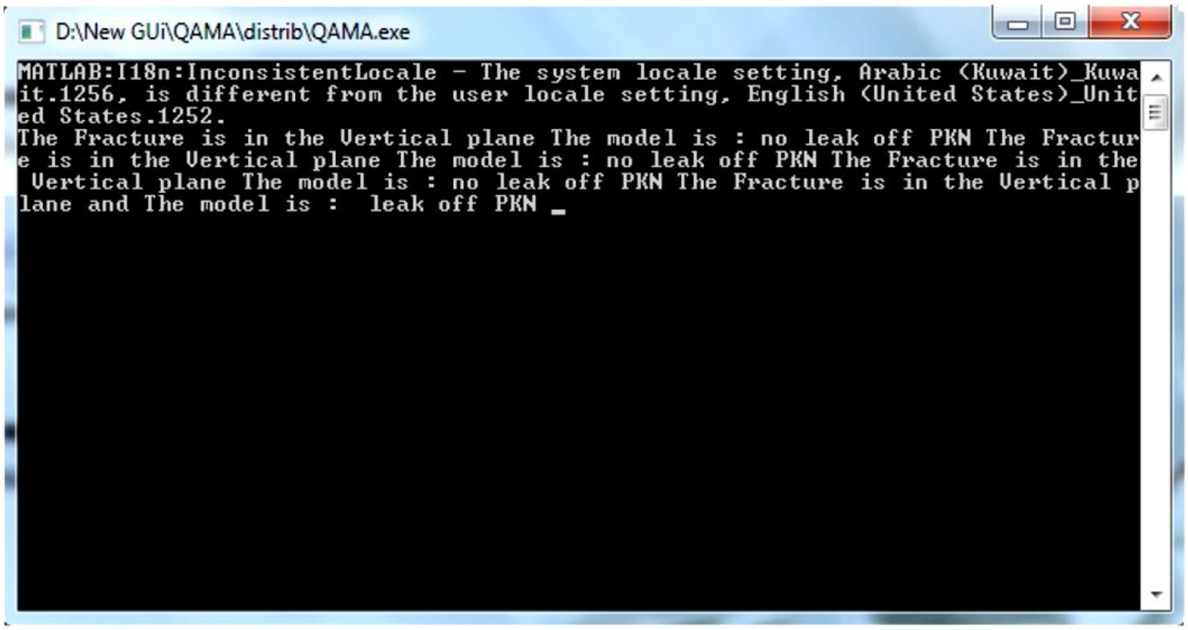


Figure (4-6) fracture shape and model being used

Note:

Young's Models change does not affect the fracture width in leak off PKN; that is for it is not exist in equation.

Chapter Five

Comments, Conclusions and Recommendations

5-1 Comments:

The results of calculation the hydraulic fracturing width using MatLab program were obtained for the following two cases: :

- The hydraulic fracturing width in the gradient stress 0.8 psi/ft and in situ stress contrast zone 50 psi comparing with the references.
- **Case (1)** the hydraulic fracturing width in the gradient stress 0.8 psi/ft and in situ stress contrast zone 50 psi for pump schedule change, flow rate equals to {40, 30, 20} bbl/min
- **Case (2):** the hydraulic fracturing width in the a gradient stress 0.8 psi/ft and in situ stress contrast zone 50 psi:
- **A:** for poisson ratio equals to {0.30, 0.25, 0.20}
- **B:** for Young's Models equals to {$1*10^6$, $1.5*10^6$, $2*10^6$}

The results for cases are compared with the references the solution in Figure 4-1 .The comparison show acceptable agreement

5-2 Conclusions:

This project presents an analytical study for all variables affective in hydraulic fracturing width calculation .

The programme was then developed and implemented on computer using MatLab.

The programme was then applied to analyze two hydraulic fracture cases.

The results obtained were compared with the results the comparison shows acceptable agreement.

The developed programme can be used to compute the hydraulic fracture width and length with simple change.

5-3 Recommendations:

For future research to calculate the hydraulic fracturing width proppant agent concentrations should be considered.

Changes to the programme , the use of the fluid pressure and pore pressure that may give more accurate results.

Developing this programme by using advance programming language to achieve a helpful simulator tool**.**

References:

1. Clifton, R.J. and Abou-Sayed, A.S.: "On the Computation of the Three-Dimensional Geometry of Hydraulic Fractures," paper SPE 7943, presented at the SPE/DOE Low Permeability Gas Reservoirs Symposium, Denver, Colorado, USA (May 20–22, 1979).
2. Craft. B. C. and Hawkins M. F.: "Applied Petroleum Reservoir Engineering. -2nd ed./revised by Ronald E. Terry ISBN 0-13-039884-5.
3. Clifton, R.J. and Abou-Sayed, A.S.: "A Variational Approach to the Prediction of the Three-Dimensional Geometry of Hydraulic Fractures," paper SPE 9879, presented at the SPE Low Permeability Symposium, Denver, Colorado, USA (May 27–29, 1981).
4. Clifton, R.J. and Wang, J-J.: "Multiple Fluids, Proppant Transport, and Thermal Effects in Three- Dimensional Simulation of Hydraulic Fracturing," paper SPE 18198, presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas, USA (October 2–5, 1988).
5. M.R.Cleary, M.Kavvades and K.Y.Lam,' Development of a Fully Three Dimensional Simulator for Analysis and Design of Hydraulic Fracturing ',SPE /DOE paper 11631,1983.
6. Bale ,A.,Oweren ,K and Smith,M.B, 'Propped Fracturing as a Tool for Sand Control and Reservoir Mangement', paper SPE 24992, also in SPE Production and facilities (1994).
7. Ahmed S. Abou – Sayed," Reservoir Stimulation". Adnantek International (1970).
8. Michael J. Economides, A. Daniel Hill,Christine Ehlig," Petroleum Production Systems ". ISBN 0-13-658683-X (1994).

9. Ching H. Yew. P . Cm,"Mechanics of Hydraulic Fracturing" .ISBN 0-88415-474-2(alk – paper) (1977) by Gulf Publishing Company.
10. Boxun Gvo, William C. Lyons, Ali Chalambor, "Petroleum Production Engineering". ISBN 075682701 (2007).
11. Robert S. Schechter,"Oil Well Stimulation". ISBN 0-13 – 949934-2 (1992).
12. Michael J. Economides, Kenneth G. Nolte, "Reservoir Stimulation". ISBN 0-13-774472-2,1989 (1989) Schlumber Educational Services
13. Michael Golan , Curtis H . Whitson "Well Performance". ISBN 0-13-946609-6 (1996) in Norway.
14. Geertsma , J. and de Klerk , F.(1969),'A Rapid Method of Predicting Width and Extent of Hydraulically Induced Fractures ,' Journal of Petroleum Technology , vol 21. no. pp. 1571-1581.
15. Nordgren ,R.P.(1972)'Propagation of a Vertical Hydraulic Fracture,' Society of Petroleum Engineers Journal, vol. 12, pp. 306-314.
16. Barenblatt, G.I.(1962),'The Mathematical Theory of Equilibrium Cracks in Brittle Materials,' Advance in Applied Mechanics, vol. 7, edited by Dryden, H.L. and von Karman, T. Academic Press, New York.
17. England, A.H. and Green, A.E.(1963),'Some Two Dimensional Punch and Crack Problems in Classical Elasticity,' Proceedings Cambridge Phil . Soc .,vol. 59, pp. 489-500.
18. Khristianovic, S. A. and Keer, L.M(1955),'Formation of Vertical Fractures by Means of Highly Viscous Liquid,' Proceedings, Fourth World Petroleum Congress, Rome, Italy, pp. 579-586
19. Abe. H. Mura, T. and Keer, L. M.(1976),'Growth Rate of a Penny-Shaped Crack in Hydraulic Fracture of Rocks,' Journal of Geophysical Research, vol. 81, no. 29, pp. 5335-5340.

22. Frasch H. Increasing the flow of oil-wells. U.S. Patent Number

556669. United States Patents and Trademark Office. Washington DC; 1896.

23. Hydraulic fracturing Operations- Well Construction and Integrity Guidelines API ,Guidance Document HF 1, First edition, October 2009.

24. Veatch RW, Moschovidis ZA, Fast CR. An overview of hydraulic fracturing. In: Gidley, Holditch, Nierode, Veatch, editors. Recent advances in hydraulic fracturing. Monograph, vol. 12. Richardson: Society of Petroleum Engineers; 1989. p. 1–38.

Appendix:

```
function varargout = QAMA(varargin)
% QAMA M-file for QAMA.fig
%      QAMA, by itself, creates a new QAMA or raises the existing
%      singleton*.
%
%      H = QAMA returns the handle to a new QAMA or the handle to
%      the existing singleton*.
%
%      QAMA('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in QAMA.M with the given input arguments.
%
%      QAMA('Property','Value',...) creates a new QAMA or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before QAMA_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to QAMA_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help QAMA

% Last Modified by GUIDE v2.5 10-Sep-2015 16:31:06
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @QAMA_OpeningFcn, ...
                   'gui_OutputFcn',  @QAMA_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before QAMA is made visible.
function QAMA_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% varargin   command line arguments to QAMA (see VARARGIN)

% Choose default command line output for QAMA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes QAMA wait for user response (see UIRESUME)
% uiwait(handles.figure1);



% --- Outputs from this function are returned to the command line.
function varargout = QAMA_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;




function vis1_Callback(hObject, eventdata, handles)
% hObject    handle to vis1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vis1 as text
%        str2double(get(hObject,'String')) returns contents of vis1 as a double



% --- Executes during object creation, after setting all properties.
function vis1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vis1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function L1_Callback(hObject, eventdata, handles)
% hObject    handle to L1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of L1 as text
```

```
%       str2double(get(hObject,'String')) returns contents of L1 as a double


% --- Executes during object creation, after setting all properties.
function L1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to L1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function ce1_Callback(hObject, eventdata, handles)
% hObject    handle to ce1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ce1 as text
%        str2double(get(hObject,'String')) returns contents of ce1 as a double
```

```
% --- Executes during object creation, after setting all properties.
function ce1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ce1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function tx1_Callback(hObject, eventdata, handles)
% hObject    handle to tx1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tx1 as text
%        str2double(get(hObject,'String')) returns contents of tx1 as a double


% --- Executes during object creation, after setting all properties.
function tx1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tx1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function den1_Callback(hObject, eventdata, handles)
% hObject    handle to den1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of den1 as text
%        str2double(get(hObject,'String')) returns contents of den1 as a double


% --- Executes during object creation, after setting all properties.
function den1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to den1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
e1=str2double(get(handles.ev,'data'));
E=e1(:,1);
v=e1(:,2);
e11=str2double(get(handles.ro,'data'));
Tm=e11(:,2);
Q=e11(:,1);
m=str2double(get(handles.vis1,'string'));
h=str2double(get(handles.L1,'string'));
Ce=str2double(get(handles.ce1,'string'));
tx=str2double(get(handles.tx1,'string'));
Rho=str2double(get(handles.den1,'string'));
figure
hold on
for i=1:length(v)
```

```
    T=[0:Tm(i)];
G(i)=E(i)/(1-v(i));
qe(i)=(2*Ce*h/sqrt(Tm(i)-tx));
if qe(i)~=0
    W=(T.^(1/8)).*((2*m*(Q(i)^2)*(1-v(i)))/((3.14^3)*h*Ce))^(1/4);
    L=(T.^(0.5)).*(Q(i)/3.14*h*Ce);
     fprintf('The Fracture is in the Vertical plane and The model is :  leak off PKN ')
else


            W=(T.^(1/3)).*((1.23)*(((8*(1-v(i))*(Q(i)^3)*m))/G(i))^(1/3));
            L=(T.^(2/3)).*((0.48*((8*G(i)*(Q(i)^3)/((1-v(i))*m)))));
            fprintf('The Fracture is in the Horizontal plane and The model is :  no leak
off KGD')


        else
            W=(T.^(1/5)).*(2.5)*(((m*(1-v(i))*Q(i)^2)/(G(i)*h))^(1/5));
            L=(T.^(4/5)).*((0.68)*((G(i)*(Q(i)^3))/((1-v(i))*m*h^4)))^(1/5);
            fprintf('The Fracture is in the Vertical plane The model is : no leak off PKN
')
end
end
plot(T,W,'-r')
 title('Time variation of Fracture Opening Width')
 xlabel('Time(min)')
 ylabel('Fracture Opening Width (in)')
 grid on
 set(gca,'GridLineStyle','-')
```

```
 a=v(1);
 b=Q(1);
 if v-a==0 & Q-b==0

text(T(length(v)),W(length(v)),['E=',num2str(E(i))],'fontname','bf','fontsize',10,'col
or','blue')
 elseif v-a==0

text(T(length(v)),W(length(v)),['Q=',num2str(Q(i))],'fontname','bf','fontsize',10,'col
or','blue')
        elseif Q-b==0

text(T(length(v)),W(length(v)),['v=',num2str(v(i))],'fontname','bf','fontsize',10,'colo
r','blue')
 end
 hold on
 end




% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ha=get(handles.ev, 'data');
ha(size(ha,1)-1,:)=[];
```

```
set(handles.ev, 'data',ha);
ha1=get(handles.ro, 'data');
ha1(size(ha1,1)-1,:)=[];
set(handles.ro, 'data',ha1);


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ha=get(handles.ev, 'data');
ha1=ha (size(ha,1),:) ;                    %last line
ha2=cat(1,ha,ha1);
set(handles.ev,'data', ha2);
ha11=get(handles.ro, 'data');
ha12=ha (size(ha11,1),:) ;                    %last line
ha22=cat(1,ha11,ha12);
set(handles.ro,'data', ha22);



% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
close



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to vis1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vis1 as text
%        str2double(get(hObject,'String')) returns contents of vis1 as a double


% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vis1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), 
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to L1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of L1 as text
%        str2double(get(hObject,'String')) returns contents of L1 as a double



% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to L1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to ce1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ce1 as text
%        str2double(get(hObject,'String')) returns contents of ce1 as a double



% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ce1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end




function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to tx1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of tx1 as text
%        str2double(get(hObject,'String')) returns contents of tx1 as a double


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tx1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to den1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of den1 as text
%        str2double(get(hObject,'String')) returns contents of den1 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to den1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```