UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

3-D NUMERICAL SIMULATION OF HORIZONTAL WELL GRAVEL PACK

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

BY

KOLAWOLE PAUL OJO

Norman, Oklahoma

2004

UMI Number: 3148881

UMI Microform 3148881

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

3-D NUMERICAL SIMULATION OF HORIZONTAL WELL GRAVEL PACK

A DISSERTATION APPROVED FOR THE

MEWBOURNE SCHOOL OF PETROLEUM AND GEOLOGICAL ENGINEERING

BY

____

Dr. Samuel Osisanya - Chairman

____

Dr. Thomas Dewers

____

Dr. Faruk Civan

____

Dr. Subhash Shah

____

Dr. Djebbar Tiab

DEDICATION

This dissertation is dedicated to my family for their love and support from the beginning of my being till present.

ACKNOWLEDGMENTS

I wish to thank the faculty and staff of the Mewbourne School of Petroleum and Geological Engineering for their cooperation, help and valuable service during my study at the University.

I would like to express my profound gratitude and appreciation to Dr. Samuel Osisanya, Chairman of my advisory committee for his supervision and assistance during this work and throughout my stay at the University. Also I am grateful to Dr. Djebbar Tiab for his invaluable advice, Dr. Subhash Shah, Dr. Faruk Civan and Dr. Thomas Dewers for their contributions and time as members of my committee.

TABLE OF CONTENTS

CHAPTER 3

CHAPTER 4

LIST OF FIGURES

Figure Number *Page*

LIST OF TABLES

ABSTRACT

A 3-D numerical simulation model for horizontal well gravel packing is developed. The model is used for simulating the gravel pack process and determining the effectiveness and pack efficiency of the gravel pack job. This model, being a true 3-D model can be used to simulate special situations in which significant leakage into the formation occurs during gravel packing. It can also be used to model recent horizontal well gravel pack techniques like Alternative Path© technology in which significant radial flow occur at different points in space and time during the process. The simulation model is particularly suitable for deep horizontal wells because the hindered settling effect prevalent in the long vertical section of deep horizontal wells is accounted for in the modeling equations.

The model was developed based on the fundamental laws of physics by solving for an overall conservation of forces during the placement of the gravel in the horizontal well bore. The equations of continuity, motion, and deposition are simultaneously solved to give a 3-D solution of the gravel pack process. The settling effect of slurry

using the type of fluid, proppant and flow conditions existing in gravel pack operations in horizontal wells were also included in the model.

The modeling equations consist of five partial differential equations with five dependent variables, all of which were solved simultaneously. The system of partial differential equations was approximated using the central-difference finite-difference approximations method. The resulting system of non-linear algebraic equations was then solved using the Newton's method of solving non-linear simultaneous equations. A solution algorithm was developed and implemented using a computer program. HGPSIM, a horizontal well gravel pack simulator was developed to implement the solution algorithm. The usefulness and validity of the model was also illustrated by simulating some horizontal gravel pack jobs reported in the literature and performing sensitivity study using the developed simulator.

The Chief Technical Contribution of this Study are;

1. A 3-D numerical simulation model is developed that offers a significant improvement over the currently available empirical, 1-D, 2-D and pseudo 3-D horizontal well gravel pack models.

2. The model is a true 3-D model and can be used to simulate special situations in which significant leakage into the formation occur during gravel packing.

3. The developed horizontal well gravel pack model can be used for simulating the gravel pack process and determining the effectiveness, total pack efficiency and average pressure throughout the well bore during the gravel pack job.

4. The simulator is useful for designing jobs where there is a slim margin between reservoir fracture pressure, equipment failure pressure, and surface injection pressure by predicting the expected surface injection pressure before committing expensive assets to the job.

C H A P T E R 1

FORMULATION OF THE PROBLEM

1.1 INTRODUCTION

Sand production from petroleum wells is one of the oldest problems facing the oil and gas industry. This problem occurs in wells producing from the younger tertiary age reservoirs, particularly in the Miocene and Pliocene age sands. Formations from this environment are weakly consolidated by a combination of argillaceous cementing materials, the interaction of inter-granular friction and in-situ stresses, capillary forces and in sometimes by the viscosity of the fluids in place. Although older and deeper formations are expected to be more consolidated, formations have been found in deep water environments that are completely unconsolidated. This is as a result of the insufficient cementing materials and a geo-pressured environment, reducing the effects of inter-granular friction and overburden stress.

In weakly consolidated formations, the stresses caused by fluids flowing into the well bore are most of the time enough to cause fine particles to be agitated. The effect caused by these particles lodging in pore throats near to the well bore, redirects the fluid flow pattern, thereby altering the direction and magnitude of the stress fields. This leads to additional particles being dislodged. Once the destabilizing forces exceed the formation strength, increased sand production will follow. Operational problems related to sand production vary from expensive sand handling problems to the complete loss of a productive zone or even the possibility of lost well control, due to eroded surface equipment.

Deep horizontal wells especially in deep water prospects require for economic success, long horizontal sections that are high rate producers, with sustained production over significant time period and are able to prevent massive sand production that tends to degrade production.

1.1.1 SAND CONTROL METHODS

There are several methods of sand control but gravel packing is generally regarded as a logical solution to sand control problem despite its technological complexities and cost. Some of the available methods include

- Increasing the shot density and size of perforations to reduce the unit fluid velocity at the well bore which may maintain it below critical levels. Poorly consolidated sands are frequently perforated with 12 or more large perforations per foot in an attempt to increase the inflow area, and thereby restrict sand production.

- Restricting the production rate will also prevent sand production by temporarily reducing the drag forces due to fluid velocity. This limits the destabilization forces to a point below the restraining forces.
- In-situ sand consolidation techniques using an artificial material like resin coated materials. This method bonds the unconsolidated sand grains together into a strong and permeable mass around the well bore. The key to success of any in-situ process relies on effective placement of the chemical throughout the entire zone.
- Mechanical control using screen, slotted liners or pre-packed screens has been used with varying degrees of success. Success is primarily dependent on the size of produced solids. The openings in a slotted liner are sized such that approximately 10% of the produced solids will bridge on the screen. These particles in turn serve as a filter for the remaining particles.

Gravel packing is the most widely used method of controlling sand production. When properly designed and executed, this method is also the most effective for controlling sand, especially in initial completions. Many studies already conducted indicated that gravel packing is preferred for reservoirs with high flow capacity where maximum performance is desired.[1, 2] Zaleski's[1] comparison of available sand control alternatives showed that sand control by gravel packing is the best in terms of performance despite also being the most expensive.

Gravel packing entails placing a granular filter in the annular space between an unconsolidated formation and a centralized slotted liner or wire-wrapped screen. Most gravel-pack treatments are performed in cased hole completions; for situations in which extremely high productivity is required, they are also performed in open hole and under-reamed completions.

1.1.2 GRAVEL PACKING IN HORIZONTAL WELLS

Gravel packing is a well completion technique used to exclude formation sand from produced fluids and extend formation longevity. It is performed by placing a gravel retention device such as a wire wrapped screen opposite the completed interval and packing gravel around the screen to create a permeable filter. Although, gravel packing is a fairly matured operation in vertical wells and slightly deviated wells, few successful jobs have been reported in deep water applications until recently[3, 4]. The very different well bore construction requirement of high production rate and extended hole stability in deep horizontal wells often makes traditional gravel packing unsuccessful. Many of these gravel packing procedures produce less than desirable results because of incomplete placement of gravel around the screen especially when the well bore is highly deviated. This has led to several studies on adapting the conventional technology to this environment.

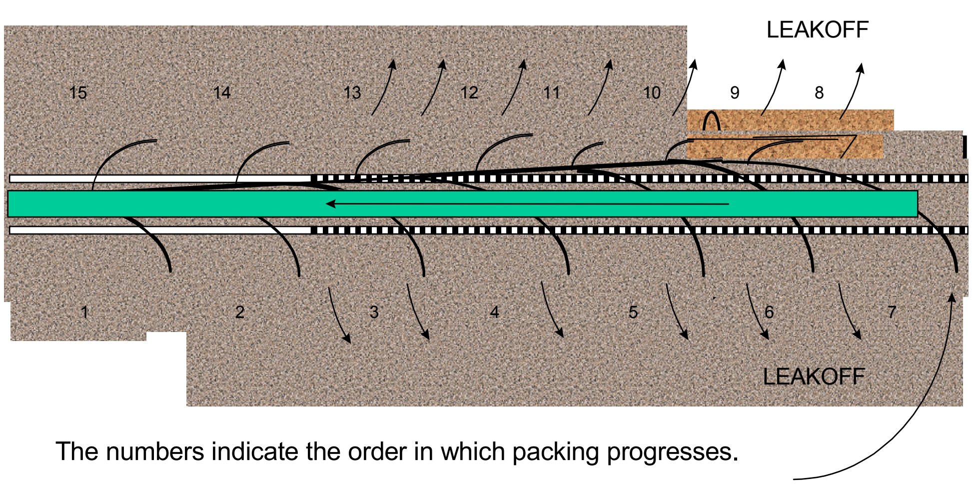
Penberthy et al. [3] provided the basis for the water packing techniques used in the majority of applications today. This technology has been successfully employed in shallow water and land-based operations on over 200 wells to date ranging from lateral lengths of a few hundred feet up to lengths exceeding 4,000 ft (1219 m) with varied results. Recently[3, 4], this technology has been applied to deepwater applications in Gulf of Mexico, West Africa and offshore Brazil. Figure 1.1 below show the gravel packing sequence in a horizontal well.

1.1.3 AVAILABLE GRAVEL PACK MODELS

Since the first series of successes were reported, drastic improvements have been recorded in the overall success rates in horizontal gravel pack in deep wells. Most of the improvements can be attributed to better operational planning, procedures and implementation in the field. Not much success have however been reported in improving numerical modeling of the overall processes involved in these operational successes. By improving the modeling of horizontal well gravel pack in deep wells, tremendous further improvements in design, implementation and evaluation is possible. This will ultimately transform into significant cost savings during well completion.

Numerous [3-7] studies have been conducted on the subject of gravel transportation and placement in order to achieve an effective gravel pack. Many of these studies are experimental, involving laboratory models while others involved developing mathematical models to simulate and design the gravel packing process before going to the field.

Figure 1.1 : Horizontal well gravel pack sequence



1.1.4 PROBLEMS WITH CURRENT MODELS

There are several problems with present models that limit their effectiveness for rigorous gravel pack designs of deep horizontal wells. One of the problems is that most of the models are simplistic and developed for slightly deviated vertical wells and therefore does not properly capture the horizontal well gravel pack process. Even the most rigorous of the models, the psuedo-3D are with rather crippling assumptions that under most circumstances common in horizontal well gravel packing are inaccurate.

Secondly, most of the models are not applicable to situations where flow exists in more than one direction. A good example is a situation where there is significant leakage into the formation or in the Alternative Path techniques in which some flow in the radial direction is expected at some point in the duration of the process.

Another shortcoming of current models is the numerical solutions employed. Because the modeling equations are mostly non-linear in nature, numerous assumptions are introduced to simplify the equations and make then tractable by simpler numerical methods. This increases the margin of error from such models. Also many of the available simulators were developed for slightly deviated wells, based on empirical equations suitable for deviated vertical well conditions. The applicability of these models to a strictly horizontal well bore has not been reported. Finally, another important problem is the failure to consider gravel transportation in long vertical well bore and any settling effects from this.

1.2 LITERATURE REVIEW

A detailed literature review on the subject of gravel transportation and placement models available for simulating gravel pack jobs in deviated and horizontal wells was undertaken and described in the following sections.

1.2.1 GRAVEL TRANSPORTATION AND PLACEMENT MODELS

Several authors have investigated the factors affecting gravel transportation and placement towards achieving an effective gravel pack and modeling the process. Gruesbeck et al. [5] performed experiments to measure pack efficiency as a function of screen parameter, fluid and gravel properties, completion configuration and angle of inclination of the well bore. They also developed a model to determine the height of the equilibrium bank formed during gravel packing of an inclined well bore. They concluded that packing efficiency increases with lower gravel concentration, lower gravel density, higher flow rate and increasing resistance to fluid flow in the tail pipe / screen annulus. They contended that by increasing the ratio of tail pipe diameter to the inside diameter of screen beyond 0.6, better efficient packing could be achieved. Several successful jobs were reportedly performed with this philosophy. Carrier fluid of viscosity less than 10 cp produced an equilibrium bank that allows complete filling of the well bore when the bank height is less than the diameter of the casing.

Hodge[6] substantiated the Gruesbeck et al. study by determining the accuracy of the predicted equilibrium bank height.

Elson, et al. [7] reported a study conducted to define optimum gravel pack procedures and completion design factors for high angle wells. Result of the study showed that high viscosity carrier fluids with high gravel concentration provides good gravel transport, but are unsuitable in wells with angles of 80^{o} from vertical. Satisfactory transport and improved packing were achieved with lower carrier fluid viscosity and sand concentrations. Tailpipe design requirements recommended by Gruesbeck et al. was used with satisfactory results.

Skaggs [8] presented the result of a large-scale well bore model he used to study gravel transport through the perforations during a high-density squeeze gravel packing operation. He concluded that the transport efficiency through perforations increases with increased fluid viscosity, gravel concentration, and annular velocity. His work is however based on vertical well bore. Winterfeld and Schroeder [9] developed a finite element numerical simulator and used it with a full-scale well bore model to study gravel placement in perforations and annulus. Their model is based on mass and momentum conservation equations.

1.2.2 HORIZONTAL WELL GRAVEL PACK PROCESS

The process of gravel placement in horizontal well has been the subject of several studies. Peden et al. [10] developed some mathematical design models for predicting the optimum combination of required design parameters such as tailpipe diameter, slurry flow rate, and gravel concentration for an optimum packing efficiency. These models were based on extensive experimental study of factors affecting packing efficiency and dimensional analysis of obtained data.

They identified from previous studies four basic flow patterns on transport of heterogeneous mixture of liquids and solids as depicted in figure 1.1. The first pattern is the symmetric suspension, which occurs at high velocities in which the solid particles are fully suspended with a symmetric concentration profile in the radial direction. The second pattern is the asymmetric suspension, which occurs at slightly, reduced velocities than symmetric suspension. The reduced slurry velocity reduced the intensity of turbulence and lift force causing a distortion of the concentration profile with more particles contained in the lower than in the upper half of the deviated borehole. The third pattern is the moving bed flow pattern. This occurs at even more reduced velocities than the latter. Here dunes settling particles are formed and they continue to move, roll and tumble along the well bore at a lower velocity than the liquid. The fourth pattern is the stationary bed saltation flow pattern. As mixture velocity is further reduced, the lowermost particles of the bed become stationary, but particle deposition continues, thus increasing the bed height and reducing the flow area. This increases the mixture velocity until particles are transported over the bed by saltation and a stationary bed saltation pattern eventually exist. This pattern according to Gruesbeck et al. [6] approximates the situation in highly deviated well bore. The application of the Peden et al. models to some design cases confirm the earlier work of Gruesbeck et al. .

1.2.3 DEVIATED WELL GRAVEL PACK MODELS

Wahlmeler and Andrews [11] in 1988 improved on the earlier works of Gruesbeck et al. and Peden et al. by developing a pseudo-three dimensional mathematical model suitable for designing and evaluating gravel pack treatments. The pseudo-three dimensional model was based on overall aspects of well bore configuration and the mechanism involved in the process of gravel packing. The model was verified both with large-scale laboratory studies and field results.

Other authors worked and developed high angle gravel packing methods and techniques. Shryock [12] worked on a full scale deviated model and had similar conclusions with earlier workers. He discovered that water carrier fluids completely gravel pack well bore deviated at 60^{o} from vertical and that packing efficiency can be improved by increasing the resistance to flow in the liner-tailpipe annulus.

Forrest [13] worked on two 45-ft and 100-ft scale models similar to that of Shryock. His experiment also confirmed mostly, the conclusions of earlier workers. In addition, he conducted flow test of gravel pack to determine if significant gravel movement occurred during simulated well production. No gravel movement was observed during single-phase production but some degree of gravel movement was observed when liquid and gas were pumped together.

Penberthy and Echols [14] looked at gravel placement techniques available using water and viscous fluids for different degree of well deviations. They concluded that water is a more general-purpose transport fluid for gravel packing than viscous fluids. According to them, viscous fluid was best suited for low well deviations and relatively short intervals.

Penberthy et al.[3] provided the basis for the water packing techniques used in the majority of applications today. They reviewed testing and field experience gained in gravel packing several long horizontal wells and gravel packed over twenty wells with intervals ranging from 600-ft to 2500-ft. They also reported the result of extensive gravel pack experiments conducted on a 1500-ft long and 4 ½-ft diameter truly field scale model. They discovered that placing gravel to the bottom of a highly deviated well requires higher pump rate, proper sizing of screen and washpipe and a slightly different completion geometry than in a vertical well.

When the proper screen and washpipe size is selected, the gravel deposition process consists of a primary or alpha wave and a secondary or beta wave. The alpha wave packs the lower section of the well from the entrance of the horizontal section until either the toe is reached or screenout occurs leaving an open void over the gravel dune. Upon reaching the bottom, a subsequent secondary or beta wave proceed in the opposite direction of the alpha wave until the interval is completely packed. The alpha wave is dominated by gravity forces, and gravel settles until an equilibrium height is reached. The reduced flow area above the gravel results in increased local velocity. As long as the local velocity remains above the critical velocity required to transport the gravel particle, the alpha wave can reach the toe of the horizontal section.

1.2.4 HORIZONTAL WELL GRAVEL PACK MODELS

Several workers have tackled the problem of developing models for gravel packing in horizontal wells ranging from empirical to full scale pseudo-3D model. Table 1.1 shows a comparison of available gravel pack models and the current model.

Gruesbeck et al. [5] derived expressions for the critical velocity based on experiments which suggested that critical velocity depends on gravel and carrier fluid densities, gravel concentration, fluid viscosity and well deviation. Brine was used as the carrier fluid in all the experiments and field operation because testing showed that a minimum fluid return rate through the washpipe produced by brine had to be achieved or the alpha wave would stall.

Brine also produced a tighter pack with less damage to the formation. Lower concentration slurry and higher pump rates both increases gravel pack length achieved as far as an optimum return rate is maintained. Size of the gravel did not appear to affect the pack length or equilibrium alpha wave height. Based on field study, Penberthy et al.[3] presented the use of pressure data as an indication of the gravel pack progress. On the treatment pressure vs. time plot, there is a separation between initial treatment pressure and the higher pressure as the alpha wave starts to propagate. When the alpha wave reaches the bottom of the well bore, the beta wave is initiated. This is also identified on the plot by an increase in pressure-time slope. A more detailed interpretation of the pressure data that goes beyond just recognizing the start and end of the different stages is necessary and possible using the developed simulator.

Nguyen et al. [15] presented the analysis of gravel packing in deviated to horizontal wells using 3-D numerical simulation. The downside of their presentation is that the modeling equations are mostly empirical without considering the fundamental physics of the problem and the varying boundary conditions. Winterfeld and Schroeder[9] developed a finite element numerical simulator for predicting gravel placement in perforations and annulus of deviated well bore. The applicability of the model to strictly horizontal well bore was not considered in the study.

Recent improvement on gravel packing of horizontal well like the Alternative Path technique have not been well modeled to simulate the overall process. Sanders et al. [16] presented the design of gravel-pack of highly deviated wells with an alternative flow concept. The presentation apart from not being especially suited for horizontal well bore also uses equations that are purely empirical and not based directly on solving the fundamental laws of physics describing the process.

Table 1.1: GRAVEL PACK MODEL COMPARISON

No.	MODEL	TYPE	FEATURES
1	Gruesbeck et al.	0-D	0-Dimensional
			Empirical Model
			Derived by dimensional analysis on Laboratory experimental data
			Estimates equilibrium velocity and height
			Cannot determine location of bridge
			Mostly for deviated and vertical wells
			Does not account for settling effect
	Penden et al.	0-D	0-Dimensional
			Empirical Model
			Derived by dimensional analysis on Laboratory experimental data
			Estimates equilibrium velocity and height
			Determine packing efficiency of perforation and annulus of deviated well
			Evaluates effects of perforation parameter, deviation angle and carrier fluid on perforation packing efficiency
			Cannot determine location of bridge
			Mostly for deviated and vertical wells
			Does not account for settling effect
3	Wahlmeier and Andrews	Pseudo-3D	Pseudo-3 Dimensional
			Numerical simulator
			Solved conservation of mass, and momentum equations
			For vertical and deviated wells
			Suitable for multiple zones, perforation intervals and fluid types
			Mostly for deviated and vertical wells
			Determine packing efficiency of perforation and annulus of deviated well
			Does not account for settling effect

Table 1.1: GRAVEL PACK MODEL COMPARISON

No.	MODEL	TYPE	FEATURES
4	Winterfeld and Schroeder	2-D	2-Dimensional
			Uses empirical relationships
			For vertical , deviated and horizontal wells
			Allow variable wellbore configuration
			Suitable for multiple fluids
			Determine packing efficiency of perforation and annulus
			Can determine location of bridge
			Does not account for settling effect
5	Nguyen et al.	3D	3-Dimensional
			Numerical simulator
			Uses empirical relationships
			For vertical , deviated and horizontal wells
			Can determine location of bridge
			Determine packing efficiency in 3D
			Suitable for multiple fluids
			Does not account for settling effect
6	Current Model	3D	3D-Dimensional
			Numerical simulator
			Solved conservation of mass, and momentum equations
			Specifically for horizontal wells
			Can determine location of bridge
			Determine packing efficiency in 3D
			Uses fluid type used in actual horizontal well gravel pack operations
			Accounts for settling effect

1.2.5 SETTLING EFFECTS

Most models developed for predicting gravel packing efficiency did not consider the hindered settling effects of the gravel as it travels from the surface through the long vertical well bore common in deep wells. This hindered settling effect of the slurry before it enters the horizontal section could be very significant depending on the operating conditions. The settling effects of the gravel will not only affect the well bore hydraulics but also the condition of the slurry that ultimately enters the horizontal section of the well bore. In other words, this effect can make concentration and other properties of the slurry that enters the horizontal section very different from that at the surface. This will ultimately make any simulation or design obtained without correction unreliable.

Several investigators have worked on the subject of particle motion in different types of Newtonian and non-Newtonian fluids especially as it affects hydraulic fracturing. Hannah and Harrington [17] and Novotny [18] measured particle-settling rates in non-Newtonian fracturing fluids. Their experimental data showed that in concentrated slurries, the particle settle more slowly because of the increase in slurry viscosity and density. Kirby and Rockefeller [19] conducted static sedimentation experiments and reported a proppant clustering phenomenon in all fracturing gel at typical proppant concentrations. This effect resulted in average static slurry settling velocity that is considerable greater than those of single particle.

Dunand and Soucemarianadin [20] studied the settling behavior of concentrated proppant suspensions and claimed that in fluids with identical single particle settling velocities, the average settling rate of a concentrated suspension in a static non-Newtonian fluid is two to three times higher than in a corresponding Newtonian fluid. Stenour [21] proposed an equation from his experimental study on sedimentation of Newtonian suspensions for correcting the settling velocity of the particles for hindered effect. Shah [22] and McMechan and Shah [23] conducted extensive experiments on settling of various particles in stagnant and moving column of fluids. They discovered that clustering occurs in linear gels at sand concentrations below 10 Ibm/gal, which result in higher settling velocities than single particle velocity. At sand concentrations above 10 Ibm/gal, hindered settling effects are dominant. Shah [24] also presented a correlation for obtaining the apparent viscosity of various fracturing fluids with changing gelling concentrations, proppant concentrations, and temperature and shear rates. Although most of the fluids used in the experiments are mostly fracturing fluids, the results however showed the extent of variations existing at differing proppant concentration and fluid rheology. These variations are expected to be more pronounced in deep horizontal applications.

Despite all these work, few data exist on the settling effects of slurries in very long vertical well bore sections of deep horizontal wells using the type of fluid, proppant and flow conditions existing in gravel pack operations. More research needs to be done on truly field scale models that would model flow in long vertical well bore sections of deep horizontal wells. The settling properties of the slurries, under different sand concentrations, different carrier fluid types / properties, and different flow conditions that are similar to those existing in real gravel pack operation needs to be investigated. The developed model proposes a method of correcting the slurry velocity for hindered settling effect before using it in the modeling equations.

1.3 RESEARCH OBJECTIVE

This study addressed some of the shortcomings in existing models for simulating gravel pack in horizontal wells. The purpose of the study is to develop a 3-D gravel pack numerical simulation model for simulating the gravel packing process and determining the effectiveness and pack efficiency of the gravel pack job in a deep horizontal well.

This model will be a true 3-D model that can be used to simulate special situations in which significant leakage into the formation occur during gravel packing. It can also be used to model recent horizontal well gravel pack techniques like Alternative Path© technology in which significant radial flow occurs at different points in space and time during the process.

The simulation model will be suitable for deep horizontal wells because the hindered settling effect prevalent in the long vertical section of deep horizontal wells will be accounted for in the modeling equations.

1.4 RESEARCH APPROACH

The model will be developed based on the fundamental laws of physics by solving the equation of conservation of forces during the placement of the gravel in the horizontal well bore. The equations of continuity, motion, and deposition will be simultaneously solved to give a 3-D solution of the gravel pack process. The settling effect of slurry using the type of fluid, proppant and flow conditions existing in gravel pack operations in the vertical and horizontal sections of deep horizontal wells will also be included in the model.

The modeling equations consist of five partial differential equations with five dependent variables, all of which will have to be solved simultaneously. Finite difference approximations to the equations, initial and boundary conditions will be derived and resulting system of non-linear equations will be solved using appropriate numerical methods.

The obtained solution algorithm will be implemented by writing a computer program with easy to use user interface and modules. HGPSIM, a horizontal well gravel packing program will be developed to implement the solution algorithm and used to simulate some horizontal gravel pack jobs reported in literature. The usefulness and validity of the model will be illustrated by simulating some horizontal gravel pack jobs reported in the literature.

1.5 ORGANIZATION OF THE DISSERTATION

The dissertation will be divided into six chapters. The current chapter detailed the formulation of the problem by introducing the problem of sand control and why gravel pack is often the preferred method. Also introduced in Chapter 1 is gravel packing in deviated and horizontal wells, available gravel pack simulation models and problem with available models. Chapter 1 also present a literature review on the subjects of gravel transportation and placement, horizontal well gravel pack methods and models and the issue of settling effects. Finally, the dissertation research objective and approached is presented. Also presented is the organization of the dissertation.

Chapter 2 deals with the formulation of 3-D numerical simulation model. The derivation of the 3-D model equations from the equations of continuity, motion and deposition is presented with the initial and boundary condition pertaining to each equation.

Chapter 3 presents the solution approach to the modeling equation of the previous chapter. The finite difference approximation to the modeling equations, and the method of solving the resulting system of non-linear algebraic equations is presented. Also presented is a solution algorithm for the system of equations.

Chapter 4 describes a computer program called HGPSIM, which was developed to implement the solution algorithm of the previous chapter. The program is a true 3-D simulator for modeling gravel packing in horizontal wells especially in deep prospects. The program's various modules and features are discussed in the chapter. The various user interfaces of the program are also presented.

Chapter 5 presents the results obtained from using the HGPSIM program to simulate three horizontal well gravel packs available in literature and performing some sensitivity study. The results from these exercises demonstrate the validity and usefulness of the developed model.

Chapter 6 presents the summary, conclusions, and recommendation of the dissertation.

C H A P T E R 2

3-D HORIZONTAL WELL GRAVEL PACK NUMERICAL SIMULATION MODEL DESCRIPTION

2.1 INTRODUCTION

The development of the gravel pack model for deep horizontal well is predicated on the fundamental laws of physics by solving for an overall conservation of forces during the placement of the gravel in the horizontal well bore. The equations of continuity, motion, and deposition are simultaneously solved to give a 3-D model of the gravel pack process. The settling effect of slurry using the type of fluid, proppant and flow conditions existing in gravel pack operations in the vertical and horizontal sections of deep horizontal wells were also included in the model.

2.2 MODEL ASSUMPTIONS

The following assumptions were made in developing the model;

- The carrier fluid is water or brine, exhibiting Newtonian fluid behavior with constant viscosity.

- The gravel and carrier fluid mixture called slurry also exhibit the property of the carrier fluid (Newtonian fluid behavior).

- The horizontal wellbore have been properly conditioned and is assumed stable throughout the gravel pack process.

- The wellbore, screen and tell-tail assembly is assumed to be cylindrical and concentrically placed in the wellbore, while the latter is isolated between the gravel pack packer and sump packer.

- The principle of conservation of mass and momentum is applicable to the flow of slurry in the horizontal wellbore/screen annulus.

- Flow exists only in the radial and horizontal direction. There is no flow in the θ direction.

- There is no mass transfer between solid and liquid phases.

- The gravel placement process follows the alpha and beta wave pattern observed in large scale laboratory experiments and field operations.

- Fluid leakoff into the formation is governed by reservoir properties, filter cake deposited from hole conditioning phase of the well completion, and condition in the horizontal wellbore during gravel deposition.

- Accurately determining fluid loss amounts into the formation is beyond the scope of this study. Fluid flow into the formation is assumed to be governed by Darcy equation.

2.3 EQUATION OF CONTINUTY

Considering the horizontal gravel pack section shown in Fig. 2.1, the continuity equation in cylindrical coordinates through the annular section is given as:

$$\frac{\partial \rho}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}\left(\rho r v_r\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\rho v_\theta\right) + \frac{\partial}{\partial z}\left(\rho v_z\right) = 0 \qquad 2.1$$

Expanding the partial derivatives of Eq. 2.1 we have

$$\frac{\partial \rho}{\partial t} + v_r \frac{\partial \rho}{\partial r} + \frac{v_\theta}{r}\frac{\partial \rho}{\partial \theta} + v_z \frac{\partial \rho}{\partial z} = -\rho\left(\frac{\partial v_r}{\partial r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{\partial v_z}{\partial z}\right) \qquad 2.2$$

For conventional horizontal well gravel pack, we can assume that there is only flow in the z direction so that $v_r = v_\theta = 0$. Eq. 3-2 becomes:

$$\frac{\partial \rho}{\partial t} + v_z \frac{\partial \rho}{\partial z} = -\rho\left(\frac{\partial v_z}{\partial z}\right) \qquad 2.3$$

In order to model horizontal well gravel pack with new technology like alternative path technology™ or when there is significant fluid leakage into the formation, we have to consider the v_r component of velocity. Eq. 2.2 then becomes:

$$\frac{\partial \rho}{\partial t} + v_r \frac{\partial \rho}{\partial r} + v_z \frac{\partial \rho}{\partial z} = -\rho\left(\frac{\partial v_r}{\partial r} + \frac{\partial v_z}{\partial z}\right) \qquad 2.4$$

The slurry density is related to carrier fluid and gravel densities as follows:

Mass of slurry = mass of carrier fluid + mass of gravel

i.e. $M_{sl} = M_{cf} + M_g$

Therefore,

$$\rho_{sl} V_{sl} = \rho_{cf} V_{cf} + \rho_g V_g \qquad \Rightarrow$$

$$\rho_{sl} = \rho_{cf} \frac{V_{cf}}{V_{sl}} + \rho_g \frac{V_g}{V_{sl}} = \rho_{cf}\left(1 - c_p\right)\frac{V_{cf}}{V_{sl}} + \rho_g c_p \quad \Rightarrow$$

$$\rho_{sl} = c_p\left(\rho_g - \rho_{cf}\right) + \rho_{cf} \qquad 2.5$$

Where V_g, V_{cf} , V_{sl} , c_p are volumes of gravel, carrier fluid, slurry and particle volume concentration respectively. The partial derivative of Eq. 2.3 can be evaluated as follows:

$$\frac{\partial \rho}{\partial t} = \left(\rho_g - \rho_{cf}\right)\frac{\partial c_p}{\partial t} \qquad 2.6$$

$$\frac{\partial \rho}{\partial z} = \left(\rho_g - \rho_{cf}\right)\frac{\partial c_p}{\partial z} \qquad 2.7$$

$$\frac{\partial \rho}{\partial r} = \left(\rho_g - \rho_{cf}\right)\frac{\partial c_p}{\partial r} \qquad 2.8$$

$$\frac{\partial \rho}{\partial \theta} = \left(\rho_g - \rho_{cf}\right)\frac{\partial c_p}{\partial \theta} \qquad 2.9$$

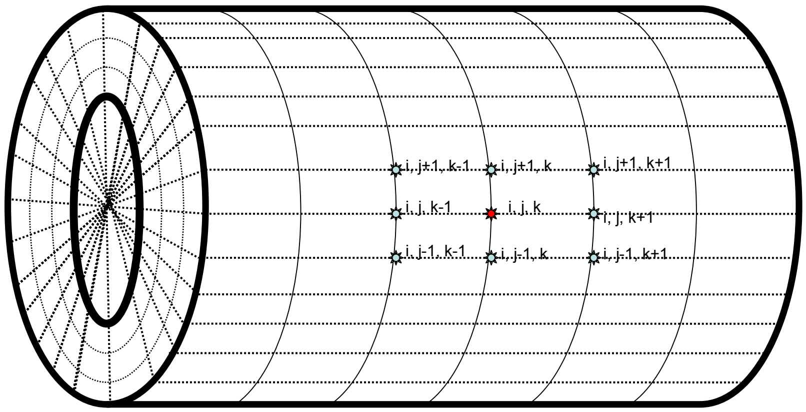


Figure 2.1 : Horizontal Gravel Pack Section

2.4 EQUATION OF MOTION

Assuming a Newtonian carrier fluid with constant density, ρ and viscosity, μ, the r, θ and z components of the equation of motion in cylindrical coordinate is respectively:

$$\rho\left(\frac{\partial v_r}{\partial t} + v_r\frac{\partial v_r}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_r}{\partial \theta} - \frac{v_\theta^2}{r} + v_z\frac{\partial v_r}{\partial z}\right) = -\frac{\partial P}{\partial r}$$

$$+\mu\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r v_r\right)\right) + \frac{1}{r^2}\frac{\partial^2 v_r}{\partial \theta^2} - \frac{2}{r^2}\frac{\partial v_\theta}{\partial \theta} + \frac{\partial^2 v_r}{\partial z^2}\right] + \rho g_r \qquad 2.10$$

$$\rho\left(\frac{\partial v_\theta}{\partial t}+v_r\frac{\partial v_\theta}{\partial r}+\frac{v_\theta}{r}\frac{\partial v_\theta}{\partial \theta}+\frac{v_r v_\theta}{r}+v_z\frac{\partial v_\theta}{\partial z}\right)=-\frac{1}{r}\frac{\partial P}{\partial \theta}$$

$$+\mu\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(rv_\theta\right)\right)+\frac{1}{r^2}\frac{\partial^2 v_\theta}{\partial \theta^2}+\frac{2}{r^2}\frac{\partial v_r}{\partial \theta}+\frac{\partial^2 v_\theta}{\partial z^2}\right]+\rho g_\theta \qquad 2.11$$

and

$$\rho\left(\frac{\partial v_z}{\partial t}+v_r\frac{\partial v_z}{\partial r}+\frac{v_\theta}{r}\frac{\partial v_z}{\partial \theta}+v_z\frac{\partial v_z}{\partial z}\right)=-\frac{\partial P}{\partial z}$$

$$+\mu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v_z}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 v_z}{\partial \theta^2}+\frac{\partial^2 v_z}{\partial z^2}\right]+\rho g_z \qquad 2.12$$

Considering that the velocity component in the θ direction, $v_\theta = 0$, then we have Eq. 2.10 through 2.12 reducing respectively to:

$$\rho\left(\frac{\partial v_r}{\partial t}+v_r\frac{\partial v_r}{\partial r}+v_z\frac{\partial v_r}{\partial z}\right)=-\frac{\partial P}{\partial r}$$

$$+\mu\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(rv_r\right)\right)+\frac{1}{r^2}\frac{\partial^2 v_r}{\partial \theta^2}+\frac{\partial^2 v_r}{\partial z^2}\right]+\rho g_r \qquad 2.13$$

$$0=-\frac{1}{r}\frac{\partial P}{\partial \theta}+\mu\left[\frac{2}{r^2}\frac{\partial v_r}{\partial \theta}\right]+\rho g_\theta \qquad 2.14$$

and

$$\rho\left(\frac{\partial v_z}{\partial t}+v_r\frac{\partial v_z}{\partial r}+v_z\frac{\partial v_z}{\partial z}\right)=-\frac{\partial P}{\partial z}$$

$$+\mu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v_z}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 v_z}{\partial \theta^2}+\frac{\partial^2 v_z}{\partial z^2}\right]+\rho g_z \qquad 2.15$$

Eq. 2.13 through 2.15 gives the partial differential equations relating the pressure variation during gravel placement with the velocity flux components.

2.5 EQUATION OF DEPOSITION

As the slurry enters each elemental volume, the rate of surface deposition of the gravel is directly proportional to the particle mass flux, the particle volume concentration in the slurry, the amount of void space in the elemental volume available for deposition and the volume available for deposition in the surrounding elements. Utilizing a form of the relationship already established by Lichtner [25] and Civan[26] that relates the rate of deposition to pores available for deposition as:

$$-\frac{\partial \varphi}{\partial t}=k_d\left(\sqrt{v_r^2+v_z^2}\right)c_p\varphi^{\frac{2}{3}} \qquad 2.16$$

k_d is a constant that accounts for two factors in the gravel deposition process. First, it serves as the constant of proportionality for the equation and secondly it serves as a constant that accounts for the volume available for deposition in the surrounding elements to the one under consideration. φ is the instantaneous elemental void space available for deposition. Figure 2.2 depicts an elemental volume schematic showing the constituent of the instantaneous elemental volume.

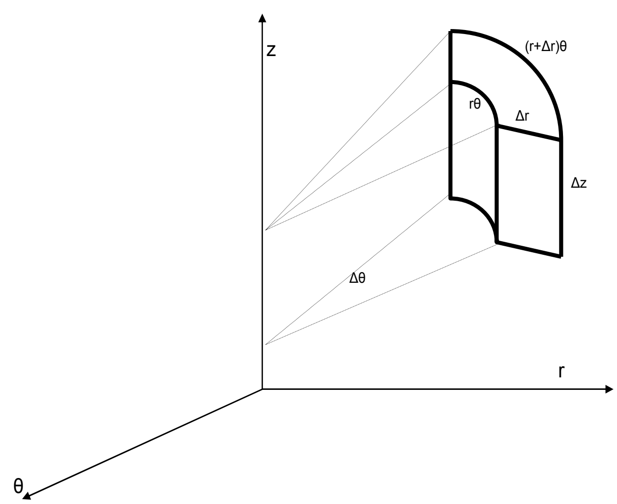


Figure 2.2: Schematic of an elemental volume

2.6 INITIAL AND BOUNDARY CONDITIONS

The initial and boundary conditions relevant to the modeling Eq. 2.1 through 2.15 is considered as follows

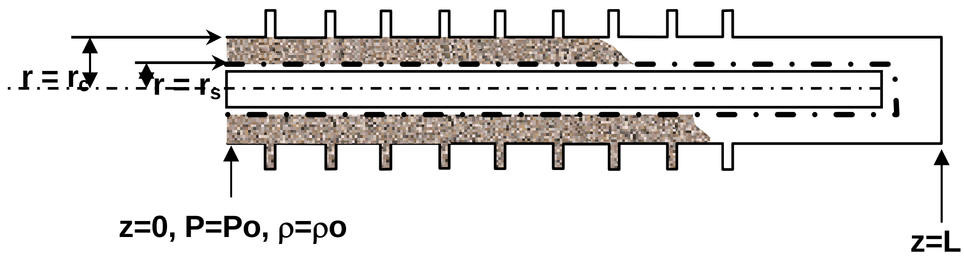


Fig. 2.3 : Schematic showing boundary and initial conditions

Continuity equation: The continuity Eq. 2.4 is subjected to the following initial condition:

$$\text{At} \quad z = 0, \quad t = 0, \quad \rho = \rho_o, \quad v_r = 0, \quad v_z = v_o \qquad 2.17$$

The boundary conditions are as follows:

$$\text{At} \quad r = r_c, \quad v_r = f(r, z, P, P_r), \quad v_z = 0 \qquad 2.18$$

and

$$\text{At} \quad r = r_s, \quad v_r = 0 \quad v_z = 0 \qquad 2.19$$

where the r component of velocity v_r is a function to be defined. This function depends on the elemental radius r, axial distance z, pressure in the elemental volume, P and reservoir pressure P_r.

Equation of motion: The equations of motion 2.13 through 2.15 are subjected to the following initial condition:

$$\text{At} \quad z = 0, \quad t = 0, \quad \rho = \rho_o, \quad v_r = 0, \quad v_z = v_o, \quad P = P_o \qquad 2.20$$

The boundary conditions are follows:

$$\text{At} \quad r = r_c, \quad v_r = f(r, z, P, P_r) \quad v_z = 0 \quad P = P_r \qquad 2.21$$

and

$$\text{At} \quad r = r_s, \quad v_r = 0 \quad v_z = 0 \qquad 2.22$$

Equation of deposition: The equation of deposition 2.16 is subjected to the following initial condition:

$$\text{At} \quad t = 0, \quad \rho = \rho_o, \quad v_r = 0, \quad v_z = v_o, \quad \varphi = 1 \qquad 2.23$$

The boundary conditions are as follows:

$$\text{At} \quad r = r_c, \quad v_r = f(r, z, P, P_r) \quad v_z = 0 \qquad 2.24$$

and

$$\text{At} \quad r = r_s, \quad v_r = 0 \quad v_z = 0 \qquad 2.25$$

2.7 SETTLING EFFECT

When a mixture of two phases that differs in density and viscosity flow in a vertical pipe, there [27] is the existence of “slip” of one phase past the other usually called “hold-up”. The phenomenon manifests in form of a change in the in-situ volume fraction or concentration of one phase as flow continues over the input or original volume fraction before flow starts. In the case of downward flow in pipes, the in-situ concentration actually decreases below the input concentration. Newitt et al. [28] studied this phenomenon and derived an equation for estimating the in-situ volume fraction of the solid in a fluid-solid system flowing downward in a vertical pipe as

$$E_s = -\left(\frac{V_m + S}{2S}\right) + \left[\left(\frac{V_m + S}{2S}\right)^2 + \frac{V_m C_s}{S}\right]^{1/2} \qquad 2.26$$

E_s is the in-situ volume fraction of the solid at any point during flow, C_s is the input or original volume fraction of the solid (this is the volume fraction of solid in the slurry at the surface), V_m is the mixture velocity and S is the average slip velocity between the phases. Utilization of this equation depends on being able to determine the slip velocity between the liquid and solid phases. According to Govier and

Aziz [27], this slip velocity has a relationship with the in-situ volume fraction, according to the equation

$$S = \frac{V_t}{1 - E_s} \qquad 2.27$$

2.8 RADIAL VELOCITY COMPONENT AT BOUNDARIES

Two types of horizontal well gravel pack are common in deep wells. One is open hole gravel pack in which the screen is placed in an open horizontal borehole and gravel is then packed in the annulus between the open hole and the screen. The second one is a cased hole gravel pack. Here the horizontal hole is cased, perforated before inserting the screen and gravel then packed in the casing-screen annulus. The later is less popular than the former due to the excessive cost and uncertainty of casing a long horizontal hole.

In either way, there could be significant leakage into the formation depending on the reservoir pressure, permeability and to some extent porosity. This possibility is accounted for in the boundary condition of section 3.5 by the term $v_r = f(r, z, P, P_r)$ in Eq. 2.18, 2.21 and 2.24. The pressure dependent non-Newtonian fluid leakoff into the formation is given as [29] :

$$v_r = \frac{n\varphi_f}{3n+1}\left(\frac{8k_f}{\varphi_f}\right)^{\frac{n+1}{2n}}\left(\frac{\nabla P}{2K}\right)^{1/n} \qquad 2.28$$

For a Newtonian carrier fluid in an elemental volume of Fig. 3.2, Eq. 2.28 reduces to:

$$v_r = \frac{k_f}{\mu}\left(\frac{P_r - P}{dr}\right) \qquad 2.29$$

where k_f is the permeability of the porous medium, P is the elemental volume pressure, P_r is reservoir pressure and dr is elemental change in radius. For open hole gravel pack with significant filter cake deposition against the borehole, k_f will be influenced more by the type of filter cake deposited than the formation.

2.9 SPECIAL FLOW SITUATION

During the gravel pack operation, incorrect gravel concentration, pump rate, or high leakoff to formation can often lead to premature bridge formation along the borehole before complete annular packing is achieved. To solve this problem, techniques have been developed to provide alternative flow path for the slurry by bypassing any premature bridge formed. To model this alternative path technology, a perforated outer shroud or liner of certain radius, r_L, greater than the screen radius is placed on the outside of the screen. The assumption is that this shroud does not influence the normal gravel transportation and placement process unless a premature bridge is formed. When this happen, the shroud diverts the flow and creates a by-pass for the slurry flow.

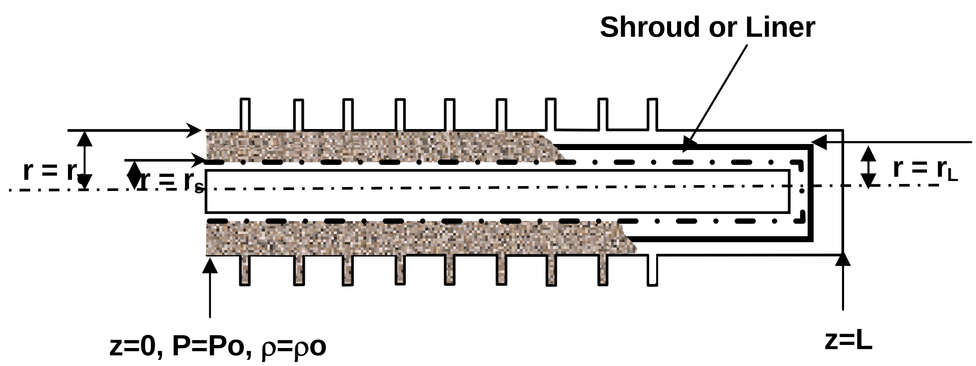


Figure 2.4: Schematic showing special flow

Under this special flow condition, the following relationships apply:

$$\forall \quad r \in \lfloor r_L, r_c \rfloor \quad \varphi = 0, \quad \Rightarrow \quad v_r = v_z \qquad 2.30$$

and

$$\forall \quad r \in \lfloor r_s, r_L \rfloor \quad \varphi = 0, \quad \Rightarrow \quad v_r = v_z \qquad 2.31$$

On the next time step, this alternative path will help to equalize the elemental pressure on both sides of the bridge, thus helping to erode the bridge, allowing normal slurry flow to resume.

2.10 FLUID LEAKOFF INTO FORMATION

The fluid leakoff rate into the formation depend on a lot of factors including the differential pressure between formation screen annulus and formation pressure, fluid transport property, formation permeability and the condition of the hole prior to gravel packing. Filter cake is deposited against the formation face to prevent fluid loss and maintain openhole stability. Determining the rate of fluid loss and formation damage resulting from such loss requires characterizing the types and properties of the drill-in fluid and completion fluid used for preparing the hole ready for gravel pack. Such characterization is necessary to be able to model the amount and property of the deposited filter cake which affects fluid leakoff rate. This characterization and subsequent estimation of fluid leakoff rate is beyond the scope of this dissertation and is part of the recommendations as an extension to this work.

2.11 SUMMARY OF EQUATIONS

To summarize the modeling equations, we have 5 system of partial differential equations in five unknowns to be solved simultaneously. The equations are as follows:

$$\frac{\partial \rho}{\partial t} + v_r \frac{\partial \rho}{\partial r} + v_z \frac{\partial \rho}{\partial z} + \rho\left(\frac{\partial v_r}{\partial r} + \frac{\partial v_z}{\partial z}\right) = 0 \qquad 2.32$$

$$\rho\left(\frac{\partial v_r}{\partial t} + v_r \frac{\partial v_r}{\partial r} + v_z \frac{\partial v_r}{\partial z}\right) + \frac{\partial P}{\partial r}$$
$$-\mu\left[\frac{\partial^2 v_r}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 v_r}{\partial \theta^2} + \frac{\partial^2 v_r}{\partial z^2}\right] - \rho g_r = 0 \qquad 2.33$$

$$\frac{1}{r}\frac{\partial P}{\partial \theta} - \mu\left[\frac{2}{r^2}\frac{\partial v_r}{\partial \theta}\right] + \rho g_\theta = 0 \qquad 2.34$$

$$\rho\left(\frac{\partial v_z}{\partial t} + v_r \frac{\partial v_z}{\partial r} + v_z \frac{\partial v_z}{\partial z}\right) + \frac{\partial P}{\partial z}$$
$$-\mu\left[\frac{\partial^2 v_z}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 v_z}{\partial \theta^2} + \frac{\partial^2 v_z}{\partial z^2}\right] - \rho g_z = 0 \qquad 2.35$$

and

$$\frac{\partial \varphi}{\partial t} = k_d\left(\sqrt{v_r^2 + v_z^2}\right)c_p \varphi^{\frac{2}{3}} \qquad 2.36$$

CHAPTER 3

SOLUTIONS TO MODELING EQUATIONS

3.1 INTRODUCTION

The equations for modeling horizontal gravel pack are derived in the previous chapter with the summary of the equations given in section 2.9 by Eq. 2.32 through 2.36. Solving the first four equations simultaneously, will give the elemental slurry density, ρ, and thus gravel concentration $c_{p.}$ Also obtainable from these first four equations are the radial and axial components of flow velocity v_r and v_z respectively, and finally the pressure distribution within each grid block. After obtaining c_p, v_r and v_z, the last equation, Eq. 2.36 is used to predict the amount of deposition occurring in each elemental volume at different time intervals by estimating the instantaneous elemental available volume for deposition φ.

This chapter details the approximation of the modeling equations using central finite different scheme and the solution of the resulting non-linear algebraic equations using the Newton method for solving non-linear equations.

3.2 FINITE DIFFERENCE APPROXIMATION

The finite difference approximation for a function U(r, θ, z, t) is given as:

$$\frac{\partial U}{\partial t} \approx \frac{U_{i,j,k}^{n+1} - U_{i,j,k}^{n}}{\Delta t} \qquad 3.1$$

$$\frac{\partial U}{\partial r} \approx \frac{U_{i+1,j,k}^{n} - U_{i-1,j,k}^{n}}{2\Delta r} \qquad 3.2$$

$$\frac{\partial^2 U}{\partial r^2} \approx \frac{U_{i+1,j,k}^{n} - 2U_{i,j,k}^{n} + U_{i-1,j,k}^{n}}{(\Delta r)^2} \qquad 3.3$$

Application of these finite difference approximations to Eq. 2.32 through 2.36 results in the following Eq. 3.4 through 3.8 respectively:

$$\left(\frac{\rho_{i,j,k}^{n+1} - \rho_{i,j,k}^{n}}{\Delta t}\right) + \left(v_{r_{i,j,k}}^{n} \frac{\rho_{i+1,j,k}^{n} - \rho_{i-1,j,k}^{n}}{2\Delta r}\right)$$
$$+ \left(v_{z_{i,j,k}}^{n} \frac{\rho_{i,j,k+1}^{n} - \rho_{i,j,k-1}^{n}}{2\Delta z}\right)$$
$$+ \left(\rho_{i,j,k}^{n} \frac{v_{r_{i+1,j,k}}^{n} - v_{r_{i-1,j,k}}^{n}}{2\Delta r}\right)$$
$$+ \left(\rho_{i,j,k}^{n} \frac{v_{z_{i,j,k+1}}^{n} - v_{z_{i,j,k-1}}^{n}}{2\Delta z}\right) = 0 \qquad 3.4$$

$$\left(\rho^n_{i,j,k}\frac{v^{n+1}_{r_{i,j,k}}-v^n_{r_{i,j,k}}}{\Delta t}\right)$$

$$+\left(\rho^n_{i,j,k}v^n_{r_{i,j,k}}\frac{v^n_{r_{i+1,j,k}}-v^n_{r_{i-1,j,k}}}{2\Delta r}\right)$$

$$+\left(\rho^n_{i,j,k}v^n_{z_{i,j,k}}\frac{v^n_{r_{i,j,k+1}}-v^n_{r_{i,j,k-1}}}{2\Delta z}\right)$$

$$+\left(\frac{P^n_{i+1,j,k}-P^n_{i-1,j,k}}{2\Delta r}\right)$$

$$-\left(\mu\frac{v^n_{r_{i+1,j,k}}-2v^n_{r_{i,j,k}}+v^n_{r_{i-1,j,k}}}{(\Delta r)^2}\right)$$

$$-\left(\frac{\mu}{r^2_{i,j,k}}\frac{v^n_{r_{i,j+1,k}}-2v^n_{r_{i,j,k}}+v^n_{r_{i,j-1,k}}}{(\Delta\theta)^2}\right)$$

$$-\left(\mu\frac{v^n_{r_{i,j,k+1}}-2v^n_{r_{i,j,k}}+v^n_{r_{i,j,k-1}}}{(\Delta z)^2}\right)-\left(\rho^n_{i,j,k}g_r\right)=0 \qquad 3.5$$

$$\left(\frac{1}{r_{i,j,k}}\frac{P^n_{i,j+1,k}-P^n_{i,j-1,k}}{2\Delta\theta}\right)$$

$$-\mu\left(\frac{2}{r^2_{i,j,k}}\frac{v^n_{r_{i,j+1,k}}-v^n_{r_{i,j-1,k}}}{2\Delta\theta}\right)-\left(\rho^n_{i,j,k}g_\theta\right)=0 \qquad 3.6$$

$$\left(\rho^{n}_{i,j,k}\frac{v_{z_{i,j,k}}^{n+1}-v_{z_{i,j,k}}^{n}}{\Delta t}\right)$$
$$+\left(\rho^{n}_{i,j,k}v_{r_{i,j,k}}^{n}\frac{v_{z_{i+1,j,k}}^{n}-v_{z_{i-1,j,k}}^{n}}{2\Delta r}\right)$$
$$+\left(\rho^{n}_{i,j,k}v_{z_{i,j,k}}^{n}\frac{v_{z_{i,j,k+1}}^{n}-v_{z_{i,j,k-1}}^{n}}{2\Delta z}\right)$$
$$+\left(\frac{P^{n}_{i,j,k-1}-P^{n}_{i,j,k-1}}{2\Delta z}\right)$$
$$-\left(\mu\frac{v_{z_{i+1,j,k}}^{n}-2v_{z_{i,j,k}}^{n}+v_{z_{i-1,j,k}}^{n}}{(\Delta r)^{2}}\right)$$
$$-\left(\frac{\mu}{r^{2}_{i,j,k}}\frac{v_{z_{i,j+1,k}}^{n}-2v_{z_{i,j,k}}^{n}+v_{z_{i,j-1,k}}^{n}}{(\Delta\theta)^{2}}\right)$$
$$-\left(\mu\frac{v_{z_{i,j,k+1}}^{n}-2v_{z_{i,j,k}}^{n}+v_{z_{i,j,k-1}}^{n}}{(\Delta z)^{2}}\right)-\left(\rho^{n}_{i,j,k}g_{z}\right)=0 \qquad 3.7$$

and

$$\frac{\varphi^{n+1}_{i,j,k}-\varphi^{n}_{i,j,k}}{\Delta t}=k_{d_{i,j,k}}^{n}\left(\sqrt{{v^{n}_{r}}^{2}_{i.j.k}+{v^{n}_{z}}^{2}}_{\;i,j,k}\right)c^{n}p_{i,j,k}{\varphi^{n}}^{\frac{2}{3}}_{i,j,k} \qquad 3.8$$

Eq. 3.4 through 3.7 are simplified further by introducing the following parameters,

$A = 2\,\Delta z \Delta\, r$ $\qquad$ $B = \Delta t \Delta z$

$C = \Delta r \Delta t$ $\qquad$ $D = 2(\Delta r)^2 (\Delta\theta)^2 (\Delta z)^2,$

$E = \Delta t \Delta r (\Delta\theta)^2 (\Delta z)^2$ $\qquad$ $F = \Delta t (\Delta r)^2 (\Delta\theta)^2 \Delta z,$

$G = 2(\Delta\theta)^2 (\Delta z)^2 \Delta t$ $\qquad$ $H = 2(\Delta r)^2 (\Delta z)^2 \Delta t$

$I = 2(\Delta r)^2 (\Delta\theta)^2 \Delta t$ and $J = 2\,(\Delta r)^2 (\Delta\theta)^2 (\Delta z)^2 \Delta t$

Applying the above parameters to the Eq. 3.4 through 3.7 gives Eq. 3.9 through 3.13 respectively:

$$\begin{aligned}
&A\rho^{n+1}_{i,j,k} - A\rho^{n}_{i,j,k} + Bv^{\;n}_{r_{i,j,k}}\left(\rho^{n}_{i+1,j,k} - \rho^{n}_{i-1,j,k}\right) \\
&+ Cv^{\;n}_{z_{i,j,k}}\left(\rho^{n}_{i,j,k+1} - \rho^{n}_{i,j,k-1}\right) + B\rho^{n}_{i,j,k}\left(v^{\;n}_{r_{i+1,j,k}} - v^{\;n}_{r_{i-1,j,k}}\right) \\
&+ C\rho^{n}_{i,j,k}\left(v^{\;n}_{z_{i,j,k+1}} - v^{\;n}_{z_{i,j,k-1}}\right) = 0
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
&D\rho^{n}_{i,j,k}\left(v^{\;n+1}_{r_{i,j,k}} - v^{\;n}_{r_{i,j,k}}\right) + E\rho^{n}_{i,j,k} v^{\;n}_{r_{i,j,k}}\left(v^{\;n}_{r_{i+1,j,k}} - v^{\;n}_{r_{i-1,j,k}}\right) \\
&+ F\rho^{n}_{i,j,k} v^{\;n}_{z_{i,j,k}}\left(v^{\;n}_{r_{i,j,k+1}} - v^{\;n}_{r_{i,j,k-1}}\right) + E\left(P^{n}_{i+1,j,k} - P^{n}_{i-1,j,k}\right) \\
&- G\mu\left(v^{\;n}_{r_{i+1,j,k}} - 2v^{\;n}_{r_{i,j,k}} + v^{\;n}_{r_{i-1,j,k}}\right) \\
&- \frac{\mu H}{r^2_{i,j,k}}\left(v^{\;n}_{r_{i,j+1,k}} - 2v^{\;n}_{r_{i,j,k}} + v^{\;n}_{r_{i,j-1,k}}\right) \\
&- I\mu\left(v^{\;n}_{r_{i,j,k+1}} - 2v^{\;n}_{r_{i,j,k}} + v^{\;n}_{r_{i,j,k-1}}\right) - J\rho^{n}_{i,j,k} g_r = 0
\end{aligned} \tag{3.10}$$

$$P^n_{i,j+1,k} - P^n_{i,j-1,k} - \frac{2\mu}{r_{i,j,k}}\left(v^{\ n}_{r_{i,j+1,k}} - v^{\ n}_{r_{i,j-1,k}}\right)$$

$$- 2\,\Delta\,\theta\, r_{i,j,k}\rho^n_{i,j,k} g_\theta = 0 \qquad 3.11$$

$$D\rho^n_{i,j,k}\left(v^{\ n+1}_{z_{i,j,k}} - v^{\ n}_{z_{i,j,k}}\right) + E\rho^n_{i,j,k} v^{\ n}_{r_{i,j,k}}\left(v^{\ n}_{z_{i+1,j,k}} - v^{\ n}_{z_{i-1,j,k}}\right)$$

$$+ F\rho^n_{i,j,k} v^{\ n}_{z_{i,j,k}}\left(v^{\ n}_{z_{i,j,k+1}} - v^{\ n}_{z_{i,j,k-1}}\right) + F\left(P^n_{i,j,k+1} - P^n_{i,j,k-1}\right)$$

$$- G\mu\left(v^{\ n}_{z_{i+1,j,k}} - 2v^{\ n}_{z_{i,j,k}} + v^{\ n}_{z_{i-1,j,k}}\right)$$

$$- \frac{H\mu}{r^2_{i,j,k}}\left(v^{\ n}_{z_{i,j+1,k}} - 2v^{\ n}_{z_{i,j,k}} + v^{\ n}_{z_{i,j-1,k}}\right)$$

$$- I\mu\left(v^{\ n}_{z_{i,j,k+1}} - 2v^{\ n}_{z_{i,j,k}} + v^{\ n}_{z_{i,j,k-1}}\right) - J\rho^n_{i,j,k} g_z = 0 \qquad 3.12$$

and

$$\frac{\varphi^{n+1}_{i,j,k} - \varphi^n_{i,j,k}}{\Delta t}$$

$$= k^{\ n}_{d_{i,j,k}}\left(\sqrt{v^{n^2}_{r\,i.j.k} + v^{n^2}_{z}}_{\ i,j,k}\right) c^{\,n} p_{i,j,k}\left(\varphi^n_{i,j,k}\right)^{2/3} \qquad 3.13$$

3.3 APPROXIMATION TO INITIAL AND BOUNDARY CONDITIONS

If we apply the finite different scheme used in the preceding section to the initial and boundary conditions presented in chapter 3, we obtain the following:

Continuity Equation: Equation 2.5, the continuity, is subjected to the following initial condition:

$$\text{At} \quad z = 0, \quad t = 0, \quad \rho = \rho_o, \quad v_r = 0, \quad v_z = v_o \Rightarrow$$
$$\rho^0_{i,j,0} = \rho_o, \quad v^0_{r i,j,0} = 0, \quad v^0_{z i,j,0} = v_o \quad \forall\, i, j \qquad 3.13$$

The boundary conditions are as follows:

$$\text{At} \quad r = r_c, \quad v_r = f(r, z, P, P_r) \quad v_z = 0 \qquad \Rightarrow$$
$$v^n_{r r_c, j, k} = f(r_c, z, P, P_r), \quad v^n_{z i,j,0} = 0 \qquad \forall\, i, j \qquad 3.14$$

and

$$\text{At} \quad r = r_s, \quad v^n_{r i,j,k} = 0 \quad v^n_{z i,j,k} = 0 \qquad 3.15$$

where the r component of velocity, v_r is a function to be defined. This function depends on the elemental radius, r, axial distance z, pressure in the elemental volume, P and the reservoir pressure P_r.

Equation of Motion: The equations of motion 2.13 through 2.15 are subjected to the following initial conditions:

$$\text{At} \quad z = 0, \quad t = 0, \quad \rho = \rho_o, \quad v_r = 0, \quad v_z = v_o, \quad P = P_o \quad \Rightarrow$$
$$\rho^0_{i,j,0} = \rho_o, \quad v^0_{r i,j,0} = 0, \quad v^0_{z i,j,0} = v_o \quad P^0_{i,j,0} = P_o \qquad 3.16$$

The boundary conditions are as follows:

$$\text{At} \quad r = r_c, \quad v_r = f(r, z, P, P_r) \quad v_z = 0 \Rightarrow$$

$$v_{r_{i,j,k}}^{n} = f(r_c, z, P, P_r), \quad v_{z_{i,j,0}}^{n} = 0 \qquad P_{r_{i,j,k}}^{n} = P_r \tag{3.17}$$

and

$$\text{At} \quad r = r_s, \quad v_r = 0 \quad v_z = 0 \quad \Rightarrow$$

$$v_{r_{i,j,k}}^{n} = 0 \quad v_{z_{i,j,k}}^{n} = 0 \tag{3.18}$$

Equation of Deposition: The equation of deposition 3.15 is subjected to the following initial conditions:

$$\text{At} \quad t = 0, \quad \rho = \rho_o, \quad v_r = 0, \quad v_z = v_o, \quad \varphi = 1 \quad \Rightarrow$$

$$\rho_{i,j,k}^{0} = \rho_o, \quad v_{r_{i,j,k}}^{0} = 0, \quad v_{z_{i,j,k}}^{0} = v_o \quad \varphi_{i,j,k}^{0} = 1 \tag{3.19}$$

The boundary conditions are as follows:

$$\text{At} \quad r = r_c, \quad v_r = f(r, z, P, P_r) \quad v_z = 0 \quad \Rightarrow$$

$$v_{r_{r_c,j,k}}^{n} = f(r_c, z, P, P_r), \quad v_{z_{i,j,0}}^{n} = 0 \tag{3.20}$$

and

$$\text{At} \quad r = r_s, \quad v_r = 0 \quad v_z = 0 \quad \Rightarrow$$

$$v_{r_{i,j,k}}^{n} = 0 \quad v_{z_{i,j,k}}^{n} = 0 \tag{3.21}$$

3.4 SOLUTION APPROACH

Equations 3.9 through 3.12 are non-linear algebraic equations. Dividing the wellbore model into coordinate system consisting of P * Q * R (PQR) grid system as depicted in Fig. 3.1 below and for each time interval considered, a system of 4PQR equations in 4PQR unknowns are obtained from the modeling Eq. 3.9 through 3.12.

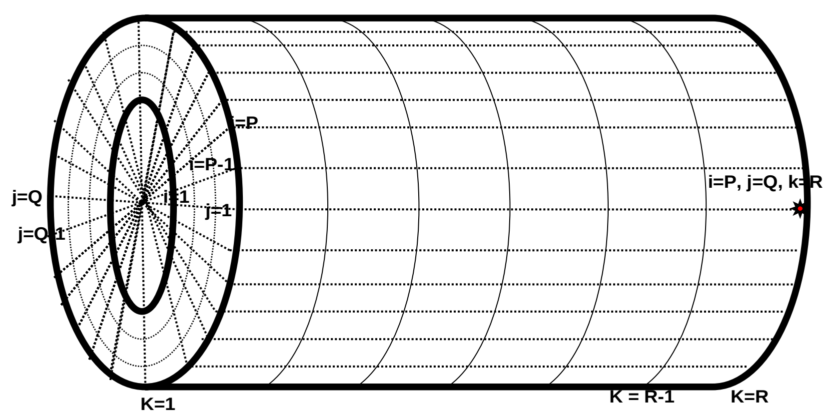


Figure 3.1 : Horizontal Gravel Pack Section showing a (P x Q x R) Grid System

The list of unknowns include $\rho_{i,j,k}$, $v_{r\,i,j,k}$, $v_{z\,i,j,k}$,and $P_{i,j,k}$ for which i= 1,2, 3, …P-1, P, j = 1,2,3……., Q-1, Q and k = 1,2,3 ……, R-1, R.

This results in a total of 4PQR unknowns. The system of 4PQR equations can be written from the system of Eq. 3.9 to 3.12 by substituting for the node points i, j, and k where i= 1,2, 3, …P-1, P, j = 1,2,3……., Q-1, Q and k = 1,2,3 ……, R-1, R.

After solving the system of Eq. 3.9 through 3.12, the obtained values of $\rho_{i,j,k}$, $v_{r\ i,j,k}$, and $v_{z\ i,j,k}$ are used to estimate for each time interval, the pore space available for deposition, $\varphi_{\ i,j,k}$ and thus the amount of gravel deposited in each grid block at that time interval.

In summary, a system of 4PQR non-linear equations in 4PQR unknowns is obtained. This system can be solved using the Newton's method of solving system of non-linear equations. The obtained result is the used to explicitly solve the last equation for the amount of gravel deposited in each grid block, within each time interval.

3.5 SOLUTION TO NON-LINEAR MODELING EQUATIONS

The numerical solution scheme employed in solving the systems of non-linear equations, Eq. 3.9 through 3.12 is the Newton's method, which is written for the system of 4PQR equations as follows [30]:

$$\begin{bmatrix} f_{1\rho_{1,1,1}} & \cdots f_{1\rho_{P,Q,R}} & \cdots f_{1vr_{P,Q,R}} & \cdots f_{1vz_{P,Q,R}} & \cdots & f_{1p_{P,Q,R}} \\ f_{2\rho_{1,1,1}} & \cdots f_{2\rho_{P,Q,R}} & \cdots f_{2vr_{P,Q,R}} & \cdots f_{2vz_{P,Q,R}} & \cdots & f_{2p_{P,Q,R}} \\ f_{3\rho_{1,1,1}} & \cdots f_{3\rho_{P,Q,R}} & \cdots f_{3vr_{P,Q,R}} & \cdots f_{3vz_{P,Q,R}} & \cdots & f_{3p_{P,Q,R}} \\ \vdots & \vdots & \cdots & \cdots \cdots f_{4vz_{P,Q,R}} & \cdots & f_{4p_{P,Q,R}} \\ \vdots & \vdots & \vdots \quad \vdots & \vdots & & \\ \vdots & \vdots & & & & \\ \vdots & \vdots & \vdots \quad \vdots & \vdots \quad \vdots & & \vdots \\ \vdots & \vdots & & & & \\ f_{4PQR\rho_{1,1,1}} & f_{4PQR\rho_{P,Q,R}} & \cdots f_{4PQRvr_{P,Q,R}} & & \cdots & f_{4PQRp_{P,Q,R}} \end{bmatrix}$$

$$X \begin{bmatrix} \Delta\rho_{1,1,1} \\ \Delta\rho_{1,1,2} \\ \Delta\rho_{P,Q,R} \\ \vdots \\ \Delta vr_{P,Q,R} \\ \vdots \\ \Delta vz_{P,Q,R} \\ \vdots \\ \Delta p_{P,Q,R} \end{bmatrix} \vdots = -\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{4PQR} \end{bmatrix} \qquad 3.22$$

Each function (f_1, f_2 .. f_{4PQR}) and partial derivative function ($f_{1\rho 1,1,1}$, $f_{1\rho 1,1,2}$, …..$f_{2\ \rho 1,1,1}$, ……….$f_{4PQR\rho P,Q,R}$) are evaluated at ($\rho_{1,1,1}$, $\rho_{1,1,2}$, ……

$\rho_{P,Q,R}$, ...$v_{r1,1,1}$, ... $v_{rP,Q,R}$, ... $v_{z1,1,1}$, ... $v_{zP,Q,R}$, ... $p_{1,1,1}$, ... $p_{P,Q,R}$), while subsequent approximation is given as follows:

$$
\begin{aligned}
&\rho_{1,1,1_{i+1}} = \rho_{1,1,1_i} + \Delta\rho_{1,1,1_i} \\
&\rho_{1,1,2_{i+1}} = \rho_{1,1,2_i} + \Delta\rho_{1,1,2_i} \\
&\vdots \\
&\rho_{P,Q,R_{i+1}} = \rho_{P,Q,R_i} + \Delta\rho_{P,Q,R_i} \\
&\vdots \\
&vr_{P,Q,R_{i+1}} = vr_{P,Q,R_i} + \Delta vr_{P,Q,R_i} \\
&\vdots \\
&vz_{P,Q,R_{i+1}} = vz_{P,Q,R_i} + \Delta vz_{P,Q,R_i} \\
&\vdots \\
&p_{P,Q,R_{i+1}} = p_{P,Q,R_i} + \Delta p_{P,Q,R_i} \qquad 3.23
\end{aligned}
$$

The method used for estimating each of the partial derivatives in Eq. 3.22 is to assume a small perturbation to each of the variables, ($\rho_{1,1,1}$, $\rho_{1,1,2}$, ... $\rho_{P,Q,R}$, ...$v_{r1,1,1}$, ... $v_{rP,Q,R}$, ... $v_{z1,1,1}$, ... $v_{zP,Q,R}$, ... $p_{1,1,1}$, ... $p_{P,Q,R}$). The partial derivatives, ($f_{1\rho1,1,1}$, $f_{1\rho1,1,2}$, ...$f_{2\ \rho1,1,1}$, ...$f_{4PQRpP,Q,R}$) are then calculated using Eq. 3.24

$$
(f_1)_{\rho_{1,1,1}} = \frac{\begin{bmatrix} f_1\left(\rho_{1,1,1}+\delta, \rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{P,Q,R},....p_{P,Q,R}\right) \\ -f_1\left(\rho_{1,1,1}, \rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{P,Q,R},....p_{P,Q,R}\right) \end{bmatrix}}{\delta}
$$

$$(f_1)_{\rho_{1,1,2}} =$$

$$\frac{\begin{bmatrix} f_1\left(\rho_{1,1,1}, \rho_{1,1,2} + \delta .....\rho_{P,Q,R} ... vr_{P,Q,R}, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \\ - f_1\left(\rho_{1,1,1}, \rho_{1,1,2} .....\rho_{P,Q,R} ... vr_{P,Q,R}, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \end{bmatrix}}{\delta},$$

$\vdots$

$\vdots$

$$\left(f_{4PQR}\right)_{\rho_{P,Q,R}} =$$

$$\frac{\begin{bmatrix} f_{4PQR}\left(\rho_{1,1,1}, \rho_{1,1,2} .....\rho_{P,Q,R} + \delta, ... vr_{P,Q,R}, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \\ - f_1\left(\rho_{1,1,1}, \rho_{1,1,2} .....\rho_{P,Q,R} ... vr_{P,Q,R}, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \end{bmatrix}}{\delta},$$

$\vdots$

$\vdots$

$$(f_1)_{vr_{1,1,1}} =$$

$$\frac{\begin{bmatrix} f_1\left(\rho_{1,1,1}, \rho_{1,1,2} .....\rho_{P,Q,R} ... vr_{1,1,1} + \delta, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \\ - f_1\left(\rho_{1,1,1}, \rho_{1,1,2} .....\rho_{P,Q,R} ... vr_{P,Q,R}, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \end{bmatrix}}{\delta}$$

$\vdots$

$$\left(f_{4PQR}\right)_{vr_{P,Q,R}} =$$

$$\frac{\begin{bmatrix} f_{4PQR}\left(\rho_{1,1,1}, \rho_{1,1,2} .....\rho_{P,Q,R}, ... vr_{P,Q,R} + \delta, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \\ - f_1\left(\rho_{1,1,1}, \rho_{1,1,2} .....\rho_{P,Q,R} ... vr_{P,Q,R}, ... vz_{P,Q,R}, ... p_{P,Q,R}\right) \end{bmatrix}}{\delta}$$

$\vdots$

$\vdots$

$$\left(f_1\right)_{vz_{1,1,1}} = \frac{\left[\begin{array}{l} f_1\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{1,1,1}+\delta,....p_{P,Q,R}\right) \\ -f_1\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{P,Q,R},....p_{P,Q,R}\right)\end{array}\right]}{\delta}$$

$\vdots$

$\vdots$

$$\vdots\left(f_{4PQR}\right)_{vz_{P,Q,R}} = \frac{\left[\begin{array}{l} f_{4PQR}\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}....vr_{P,Q,R},...vz_{P,Q,R}+\delta,....p_{P,Q,R}\right) \\ -f_1\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{P,Q,R},....p_{P,Q,R}\right)\end{array}\right]}{\delta},$$

$\vdots$

$$\left(f_1\right)_{p_{1,1,1}} = \frac{\left[\begin{array}{l} f_1\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{P,Q,R},....p_{P,Q,R}\right) \\ -f_1\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{P,Q,R},...p_{1,1,1}+\delta....p_{P,Q,R}\right)\end{array}\right]}{\delta}$$

$\vdots$

$$\left(f_{4PQR}\right)_{p_{P,Q,R}} = \frac{\left[\begin{array}{l} f_{4PQR}\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}....vr_{P,Q,R},...vz_{P,Q,R},....p_{P,Q,R}+\delta\right) \\ -f_1\left(\rho_{1,1,1},\rho_{1,1,2}.....\rho_{P,Q,R}...vr_{P,Q,R},...vz_{P,Q,R},....p_{P,Q,R}\right)\end{array}\right]}{\delta}$$

3.24

By completely solving these systems of equations at each time interval, gravel transportation and deposition process for that time period is modeled.

To solve the system of matrices of Eq. 3.22 for each time interval, the Gaussian elimination method with partial pivoting is employed. The coefficient matrix is first Lower and Upper (LU) decomposed and then solved for each time interval to give a solution matrix of improved values of the required dependent variables which are then plugged back into the matrix continuously until convergence is obtained for that time interval.

After obtaining convergence, the obtained values of the variables, $\rho_{i,j,k}$, $v_{r\ i,j,k}$, and $v_{z\ i,j,k}$ are used to estimate for each time interval, the pore space available for deposition, $\varphi_{\ i,j,k}$ and thus the amount of gravel deposited in each grid block at each time interval is estimated.

This solution algorithm is implemented in a computer program as part of the developed horizontal well gravel pack simulator (HGPSIM) presented in the following chapter.

3.6 APPROXIMATIONS TO INITIAL CONDITIONS

The initial conditions required to obtain a complete solution of the modeling equations include the variables, v_{zo}, and P_o. In order to determine the values of the initial velocities, v_{zo} in each grid block, the analytical solution of the steady state flow of an incompressible fluid in an annulus [29] is used:

$$v_{zo} = \frac{\left(P_o - P_{i,j,k}\right)R^2}{4\mu\, dz}\left[1 - \left(\frac{r_{i,j,k}}{R}\right)^2 + \left(\frac{1-k^2}{\ln\left(1/k\right)}\right)\ln\frac{r_{i,j,k}}{R}\right] \qquad 3.25$$

Where R is the open hole or casing radius, $k = r_s/R$, r_s is the screen outer radius, and $r_{i,j,k}$ is the radius of the grid block under consideration.

To determine the value P_o, consider the pressure balance in the system prior to the transportation of gravel into the horizontal section as shown in Fig. 3.2 below. P_o is estimated from Eq. 3.26 as

$$P_o = P_{injection} + P_{hydrostatic} - \Delta P_{friction} \qquad 3.26$$

The starting solution point for the initial pressure in each of the grid block can be estimated from the analytical solution of the volume rate of flow of the steady state flow of an incompressible fluid in an annulus as given by Eq. 3.27 [29] as follows:

$$P_{i,j,k} = P_o - \frac{8Q\mu\, dz}{\pi R^4 \left[\left(1-k^4\right) - \left(\frac{\left(1-k^2\right)^2}{\ln\left(1/k\right)} \right) \right]} \qquad 3.27$$

Q is the average volumetric flow rate.

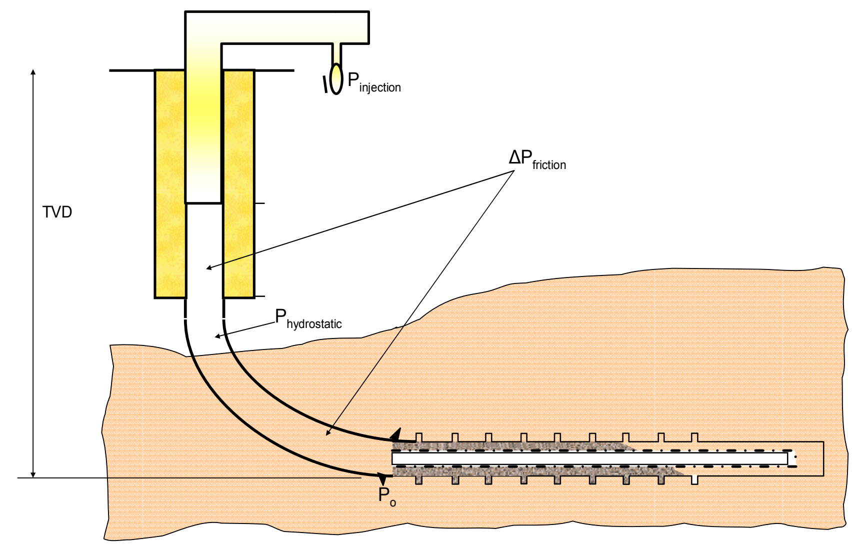


Fig. 3.2 : Schematic showing estimation of initial Pressure Po

3.7 SETTLING EFFECT SOLUTION

The equations that account for the settling effect of the slurry are given in Chapter 2 by Eq. 2.26 and 2.27. Substituting Eq. 2.27 into 2.26 and rearranging gives the following equation:

$$E_s = -\left(\frac{(1-E_s)V_m}{2V_t} + \frac{1}{2}\right) + \left[\left(\frac{(1-E_s)V_m}{2V_t} + \frac{1}{2}\right)^2 + \frac{(1-E_s)V_m C_s}{V_t}\right]^{1/2} \quad 3.28$$

Eq. 3.28 can be numerically solved to obtain the in-situ volume concentration E_s, given the input volume concentration C_s, mixture velocity, v_m and the terminal or free gravel settling velocity, v_t. The mixture velocity is taken at the entrance into the horizontal section.

A particle falling under the action of gravity will accelerate until the drag force balance gravitational force after which it falls at a constant terminal or free settling velocity[33] given by

$$V_t = \sqrt{\frac{2g_c m_g d_g\left(\rho_g - \rho_{cf}\right)}{\rho_g A_g \rho_{cf} C_D}}$$

Where m_g is mass of particle and C_D is drag coefficient, a function of Reynolds number, which is also a function of viscosity. In the Newton's law regime for spherical particles where gravel pack flows belong[10], Reynolds number is between 1000 and 350,000, and C_D = 0.445 within 13 percent [33]. The terminal settling velocity equation reduces to [33]

$$V_t = 1.74\sqrt{\frac{g_c d_g\left(\rho_g - \rho_{cf}\right)}{\rho_{cf}}} \qquad 3.29$$

The subscript cf in Eq. 3.29 represents carrier fluid while g represents gravel.

3.8 ESTIMATION OF PACKING EFFICIENCY

The packing efficiency as the gravel packing progresses can be approximated by calculating the arithmetic average of the available space for deposition during each time step of the simulation. The packing efficiency for a time step, n, can be estimated from Eq. 3.30 as follows

$$P.E = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{P} \varphi_{i,j,k}^{n}}{MxNxP} \qquad 3.30$$

The overall packing efficiency for the gravel pack job will be estimated at a time after the pumping has stopped from Eq. 3.30.

Similarly, the pack height or variation of the packing efficiency with axial distance in the wellbore can be estimated using Eq. 3.31,

$$P.E_K = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \varphi_{i,j,k}^{n}}{MxN} \qquad 3.31$$

For each axial distance k = 1, 2, …..,P at every time interval during the simulation.

3.9 ESTIMATION OF AVERAGE PRESSURE

The average pressure in the horizontal section during each time interval is estimated using the volumetric average of pressures in all grid blocks as given by Eq. 3.32,

$$P_{avg} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{P} V_{i,j,k} P_{i,j,k}^{n}}{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{P} V_{i,j,k}} \qquad 3.32$$

Similarly, the variation of average pressure with axial distance during the simulation can be determine using the Eq. 3.33 below,

$$P_{avg_K} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} V_{i,j,k} P_{i,j,k}^{n}}{\sum_{i=1}^{M} \sum_{j=1}^{N} V_{i,j,k}} \qquad 3.33$$

The elemental volume, $V_{i,j,k}$ in each of the above equation is estimated using Eq. 3.34,

$$V_{i,j,k} = \Delta r \, \Delta z \left[r\theta + \frac{1}{2} \Delta r \, \theta \right] \qquad 3.34$$

3.10 SETTLING FACTOR

The settling factor is defined to quantify the amount of correction necessary to the gravel concentration entering the horizontal wellbore before being inputted and used in the simulator. The settling factor is defined by Eq. 3.35

$$S.F. = \frac{c_p - E_s}{c_p} x100 \, \% \qquad 3.35$$

Where cp is the surface gravel concentration and Es is the gravel concentrated at input into the horizontal section corrected for settling effects using Eq. 3.28.

CHAPTER 4

HORIZONTAL WELL GRAVEL PACK SIMULATOR

4.1 INTRODUCTION

This section describes the horizontal well gravel pack simulator HGPSIM, which is a computer program developed to implement the solution schemes described in Chapter 3. The program consists of different modules for inputting the project parameter, formation parameter, the well data including data for the vertical well section and the horizontal well section, and treatment data, including data for the carrier fluid, gravel and slurry.

Also included in the simulator is the module on gravel transportation and deposition that shows the progress of the simulation. This module also shows an animated picture of how gravel is transported and deposited in the horizontal wellbore at every time step in the simulation. This module has a check box that can be used to indicate whether settling effect should be considered in the simulation or not.

Finally, the program has result output and charting module for writing the result of the simulation to an ASCI text file and then displaying the charts of some of the results of the simulation run. The various plots available include the packing efficiency vs. length plot, and average pressure versus length plot for every time step simulated. Also available is the overall job packing efficiency vs. time and average pressure vs. time plots. The following sections consider each of the modules in more detail with figures showing screen captures from the HGPSIM program.

4.2 DATA INPUT MODULE

The data input module of the HGPSIM is shown in Fig. 4.1 through 4.4. The first data entry screen, Fig. 4.1, shows the project information screen where information about the project such as client name, location, project title and date is inputted. It's important to enter a reasonable value for the project title because this is used to name the simulation result output file that is generated at the end of the simulation run. The second screen, Fig. 4.2, is the formation data input screen. Here information about the formation relevant to the gravel pack simulation process must be entered. These include the reservoir porosity, permeability, fracture gradient, reservoir pressure, etc. The third data entry screen is the shown in Fig. 4.3. This is the well data entry screen, and it's divided into two sections. The vertical wellbore section where data pertaining to the vertical wellbore is entered and the horizontal wellbore section, for entering data about the horizontal wellbore to be gravel packed. Required data for the former include, vertical section total vertical depth (TVD), vertical section measured depth (MD), last casing size and open hole diameter. The data for the horizontal wellbore include, hole or casing diameter, screen outer and inner diameters, screen length, tell-tail diameter and length, wash pipe inner and outer diameters. Finally, the fourth data entry screen is the treatment data screen shown in Fig. 4.4. This is where all data relating to the treatment carrier fluid, gravel, and slurry are entered. The injection pressure, rate and approximate volume to ascertain the approximate duration of the gravel pack job are also entered here.

4.3 GRAVEL TRANSPORTATION AND DEPOSITION MODULE

The gravel transportation and deposition module main screen is shown in Fig. 4.5. This is where the simulation process is completed by dividing the total treatment time into time intervals (minutes). The solution routine detailed in Chapter 3 is performed on each time interval. As the simulation for each interval is completed, gravel transportation and deposition module main screen is updated to reflect the wellbore condition and gives an animated view of the wellbore status. This is illustrated by Figures 4.6 and 4.7. The figures show the gravel transportation and deposition module screen status for time intervals 3 and 4 respectfully during a certain simulation run. The gravel transportation and deposition module screen shows the progress of the simulator and the overall packing efficiency achieved at the current simulation time in real time. This will allow the real time monitoring of the performance of the simulation job. Finally the screen allows the settling effects to be included in the simulation run by checking the hindered settling check box as shown in Fig. 4.5. This is particularly useful for comparing the results obtain from neglecting settling effect as it is currently being done in existing models to that of considering settling effect that is proposed in this study.

4.4 SIMULATION RESULT MODULE

The simulation result module of the HGPSIM program has the capability to generate after the simulation run, a text file containing all the result of the simulation run including the calculated velocity profiles, pressure and slurry density for each grid cell for every time step simulated. Also included in the result file is the estimated available space for gravel deposition for each grid block at every time step, estimated pack efficiency vs. length and the final pack efficiency for the simulation. Also saved is the calculated setting effect factor for the simulation run. The simulation result module also has charting capability for displaying charts of some of the result of the simulation run. One of the plots available is the packing efficiency vs. length plot, and average pressure versus length plot for every time step simulated as shown in Figures 4.8 , 4.9 and 4.10 for time duration of 27, 36 and 90 minutes respectively. Also available is the overall job packing efficiency vs. time and average pressure vs. time plot as illustrated in Fig. 4.11.

4.5 HGPSIM DESIGN AND IMPLEMENTATION

This section details the design and implementation of the HGPSIM program for simulating gravel packing in horizontal wells. The detail listing of the program is provided in Appendix A of this dissertation. The horizontal wellbore is divided into coordinate system consisting of P * Q * R grid system as depicted in Fig. 3.1. Also, the job total duration is divided into time steps so that for each time interval, a system of 4PQR Equations in 4PQR unknowns as represented by functions f1, f2, f3 and f4 respectively in the listing of Appendix A are obtained from the modeling Eq. 3.9 through 3.12.

The list of unknowns to be determined in each grid block include $\rho_{i,j,k}$, $v_{r\,i,j,k}$, $v_{z\,i,j,k}$,and $P_{i,j,k}$ for which i= 1,2, 3, …P-1, P, j = 1,2,3……., Q-1, Q and k = 1,2,3 ……, R-1, R and they are represented by rho(n,i,j,k), vr(n,i,j,k), vz(n,i,j,k) and P(n,i,j,k) respectively in Appendix A, where n represents the time interval under consideration. This results in a total of 4PQR unknowns. A total of 4PQR equations can also be written from the system of Eq. 3.9 to Eq. 3.12 by substituting for the node points i, j, and k where i= 1,2, 3, …P-1, P, j = 1,2,3……., Q-1, Q and k = 1,2,3 ……, R-1, R. The function evaldelta(n) of appendix A implements the estimation and setup of the defining matrix system of Eq. 3.22

The matrix system solution is implemented in the subroutine solvmat of Appendix A by using Gaussian elimination method with partial pivoting. The coefficient matrix is first LU decomposed and then solved for each time interval to give a solution matrix of improved values of the required variables. The improved values are then plugged back into the matrix continuously until convergence is obtained. After solving the system of Eq. 3.9 through Eq. 3.12, the obtained values of $\rho_{i,j,k}$, $v_{r\ i,j,k}$, and $v_{z\ i,j,k}$ are then used to estimate for each time interval, the pore space available for deposition, $\varphi_{\ i,j,k}$ and thus the amount of gravel deposited in each grid block at each time interval. The packing efficiencies at each time interval using Eq. 3.30 and Eq. 3.31 and finally the average pressures using Eq. 3.32 and Eq. 3.33 are estimated.

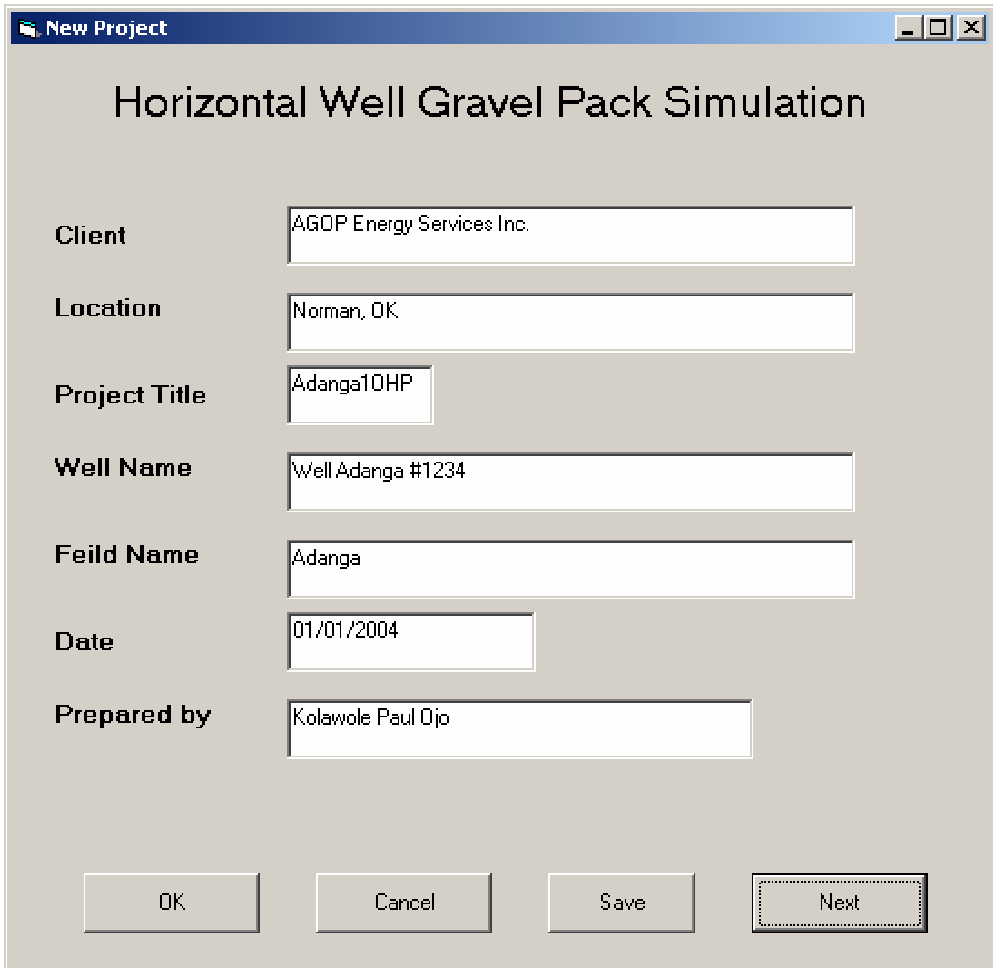


Figure 4.1: HGPSIM Project Data Input Screen

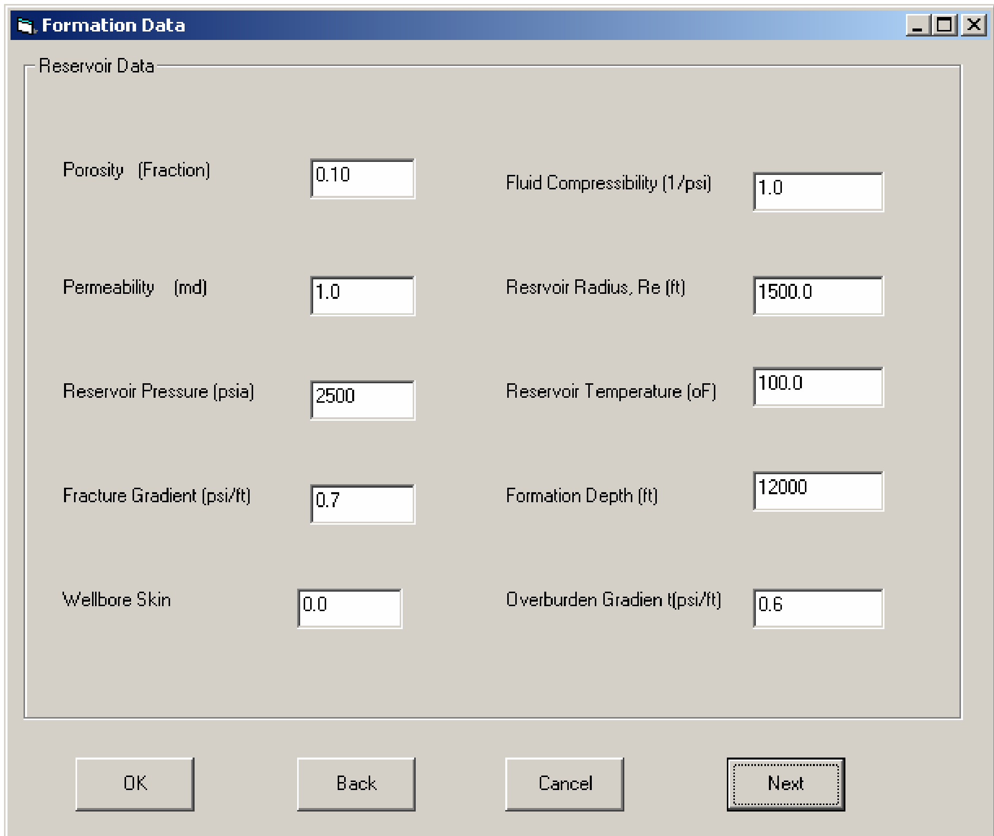


Figure 4.2: HGPSIM Formation Data Input Screen

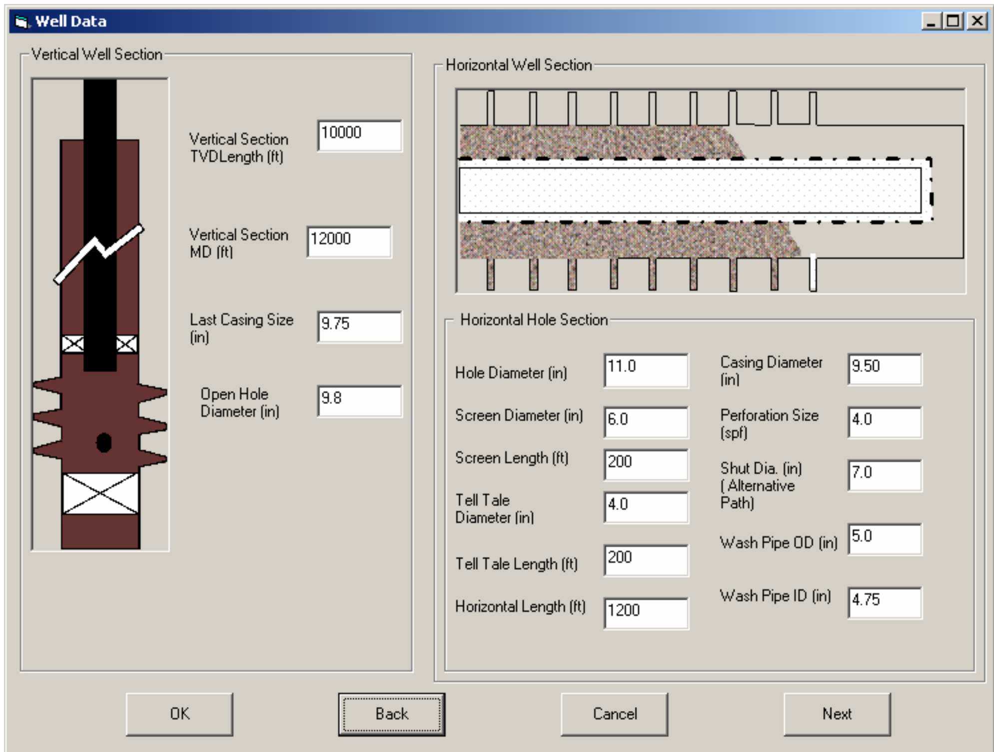


Figure 4.3: HGPSIM Well Data Input Screen

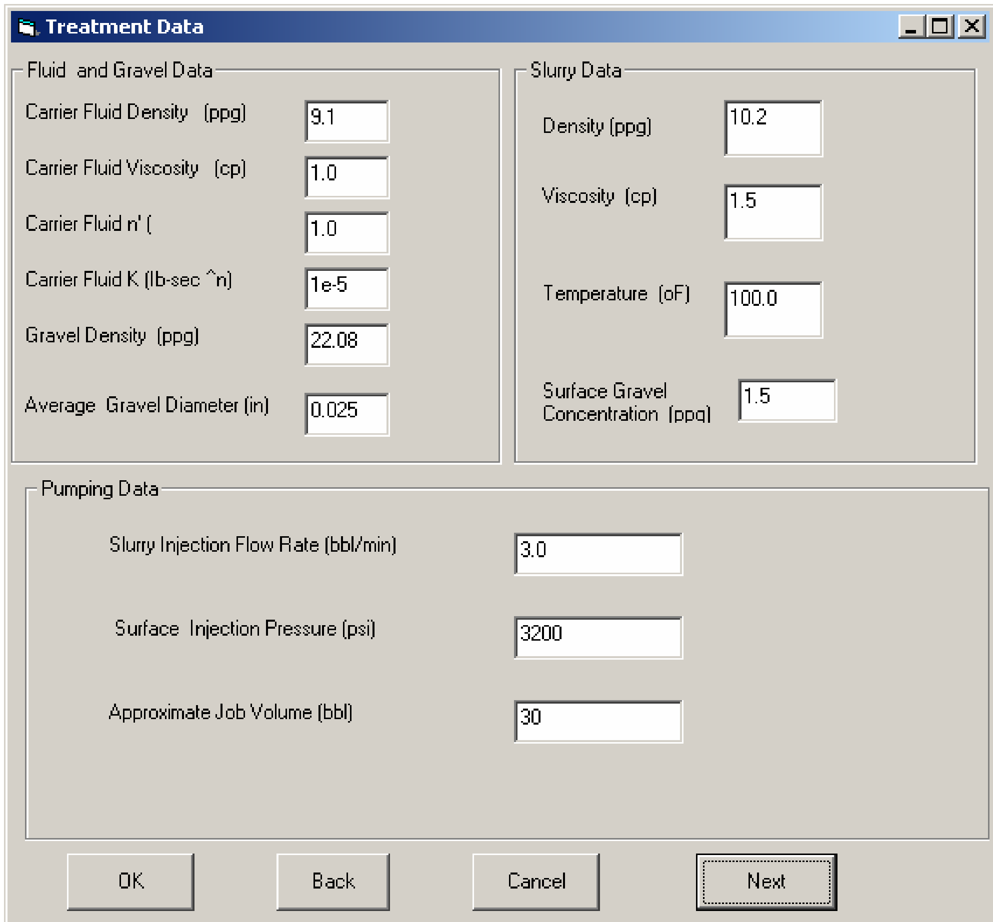


Figure 4.4: HGPSIM Treatment Data Input Screen

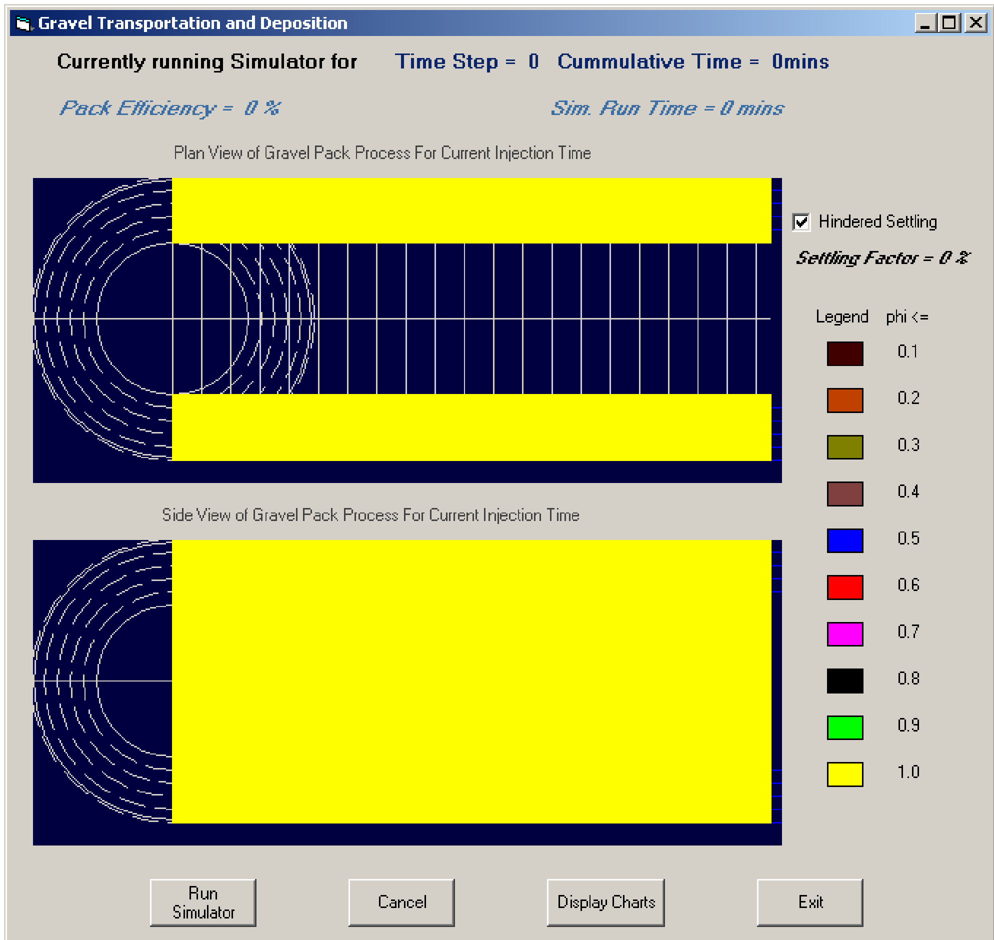


Figure 4.5: HGPSIM Gravel Transportation and Deposition Module

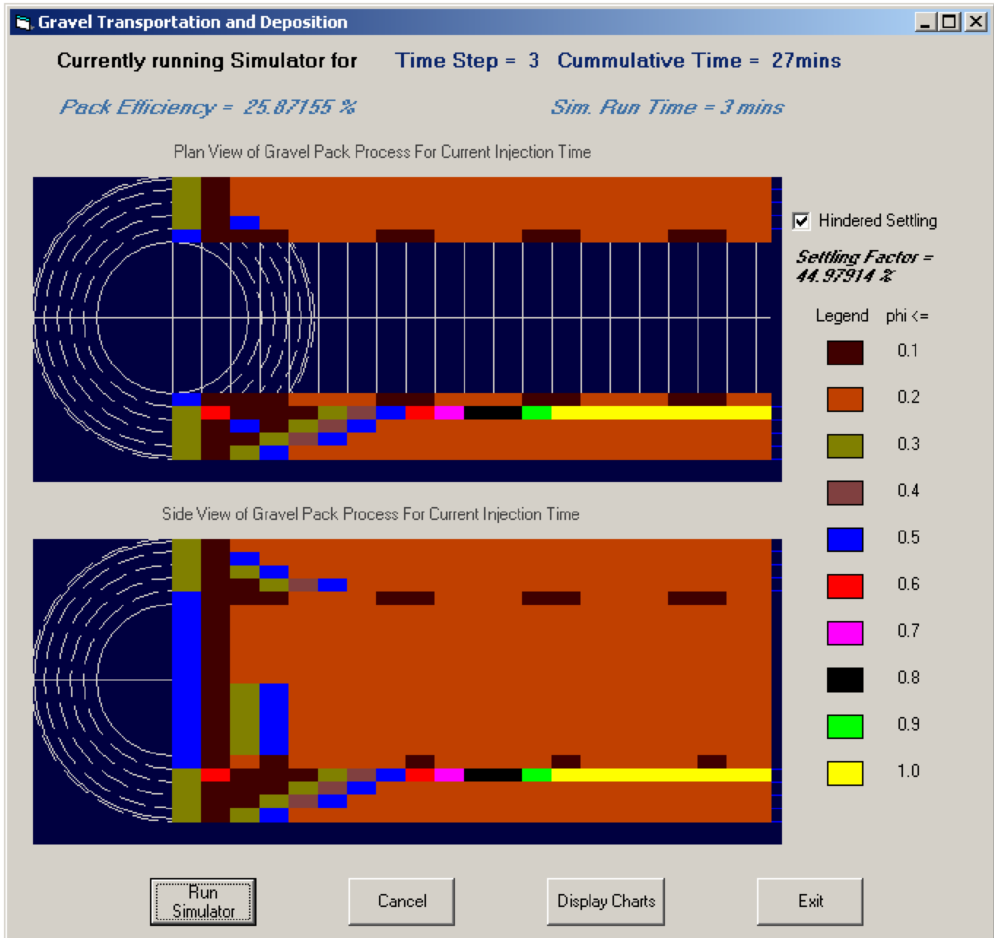


Figure 4.6: HGPSIM Simulation in Progress for Time Step = 3

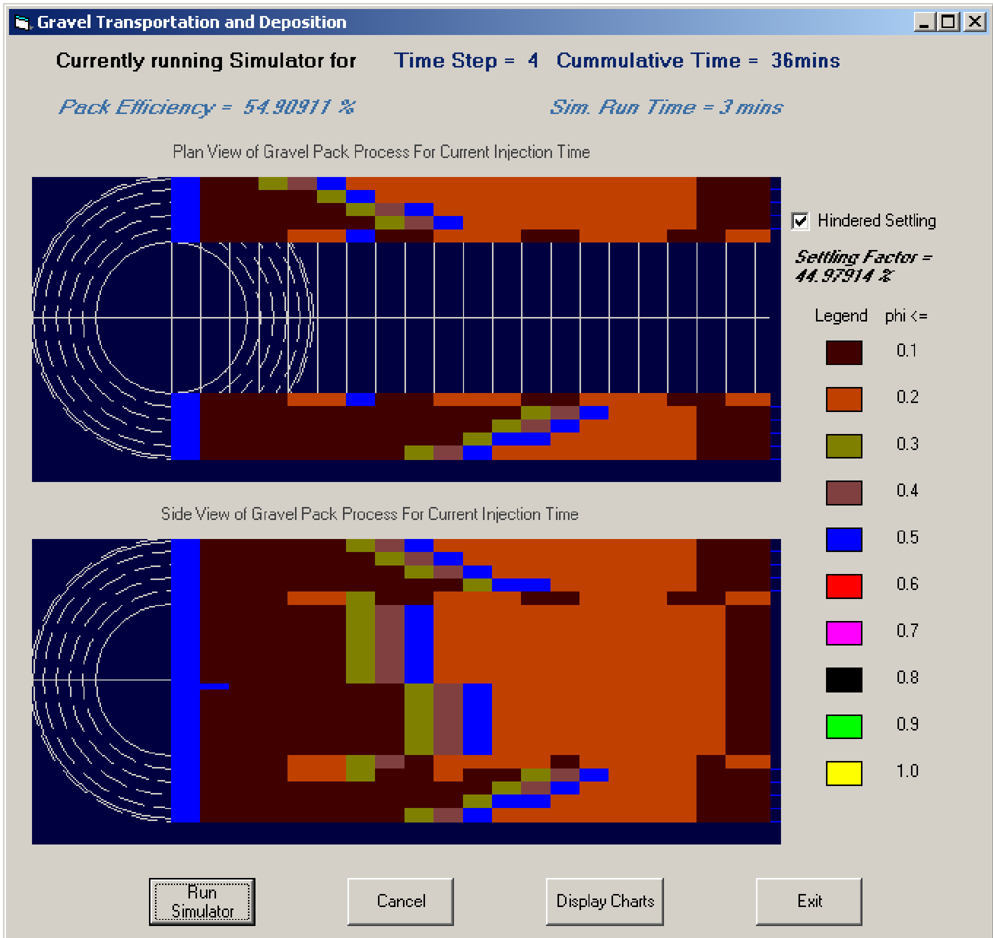


Figure 4.7: HGPSIM Simulation in Progress for time step = 4

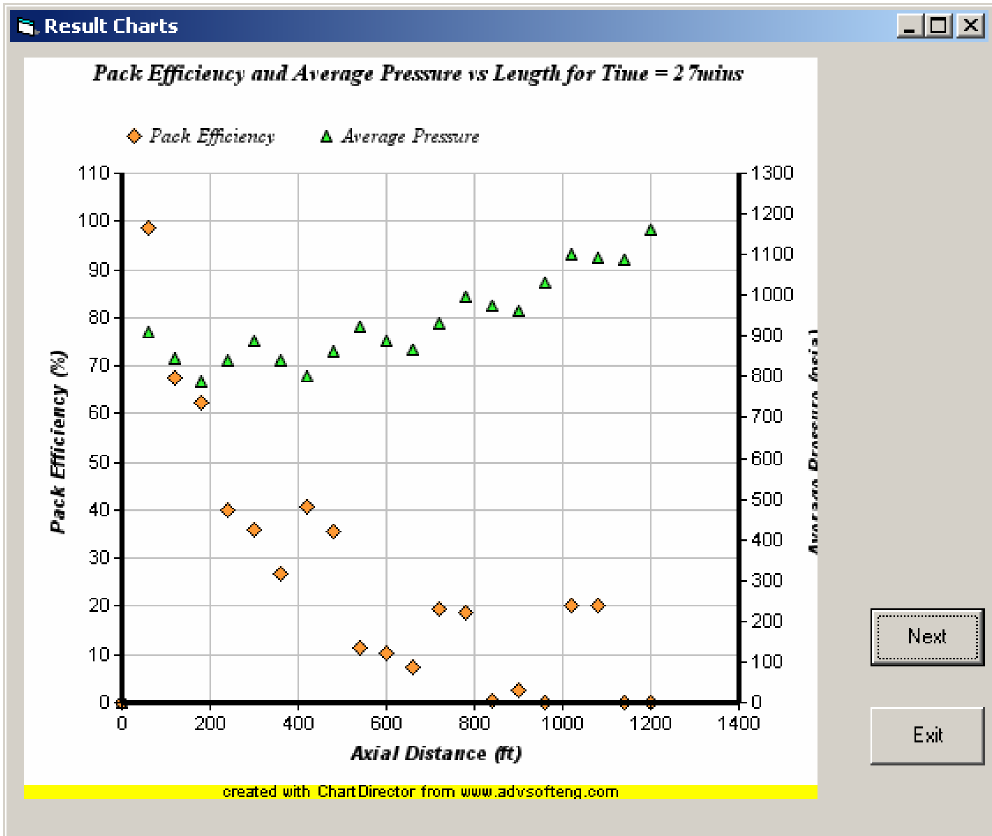


Figure 4.8: HGPSIM Result Charts Screen at Time = 27min

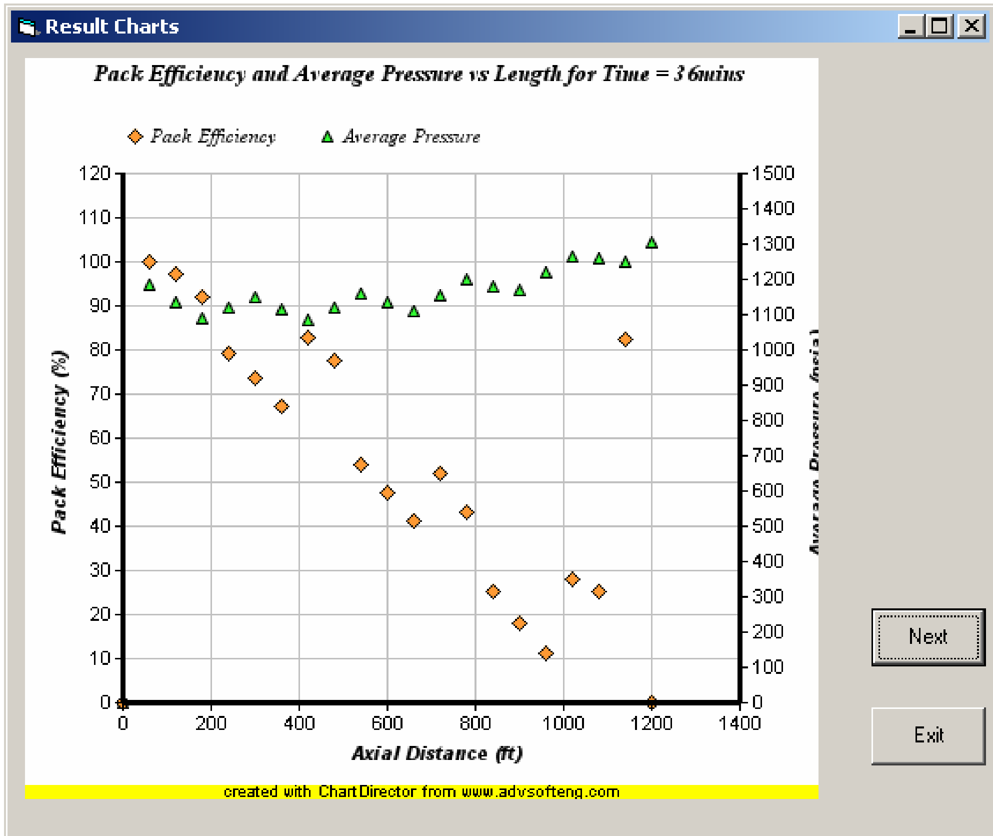


Figure 4.9: HGPSIM Result Charts Screen at Time = 36min

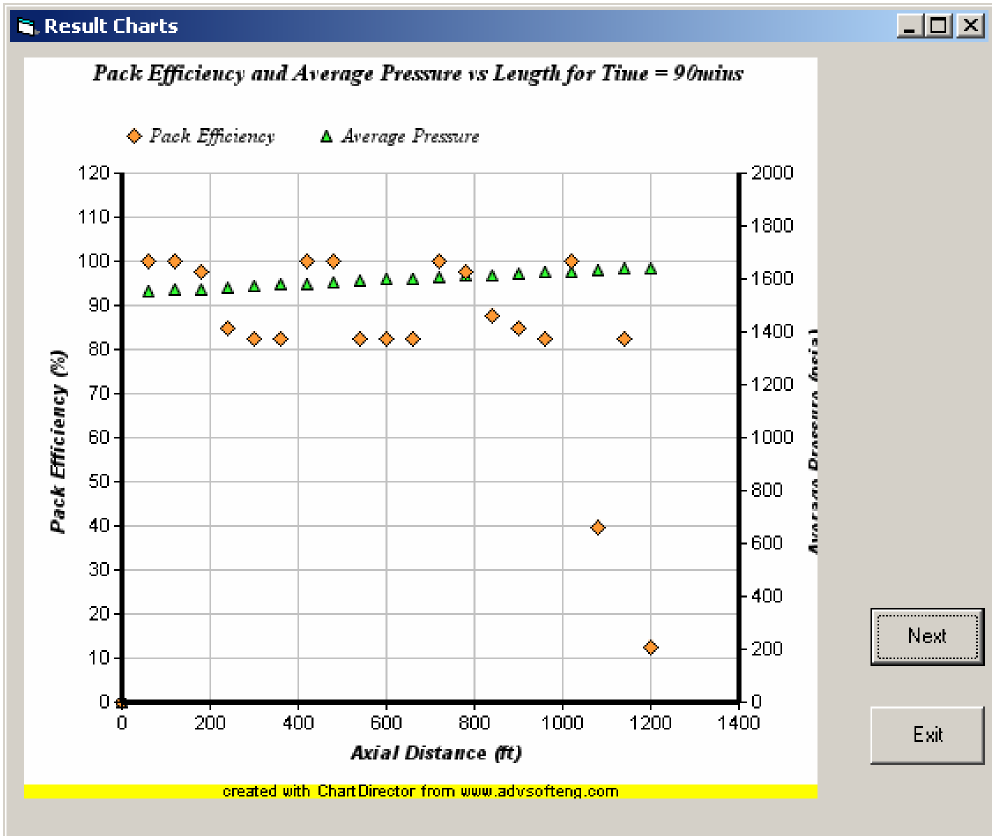


Figure 4.10: HGPSIM Result Charts Screen at Time = 90mins

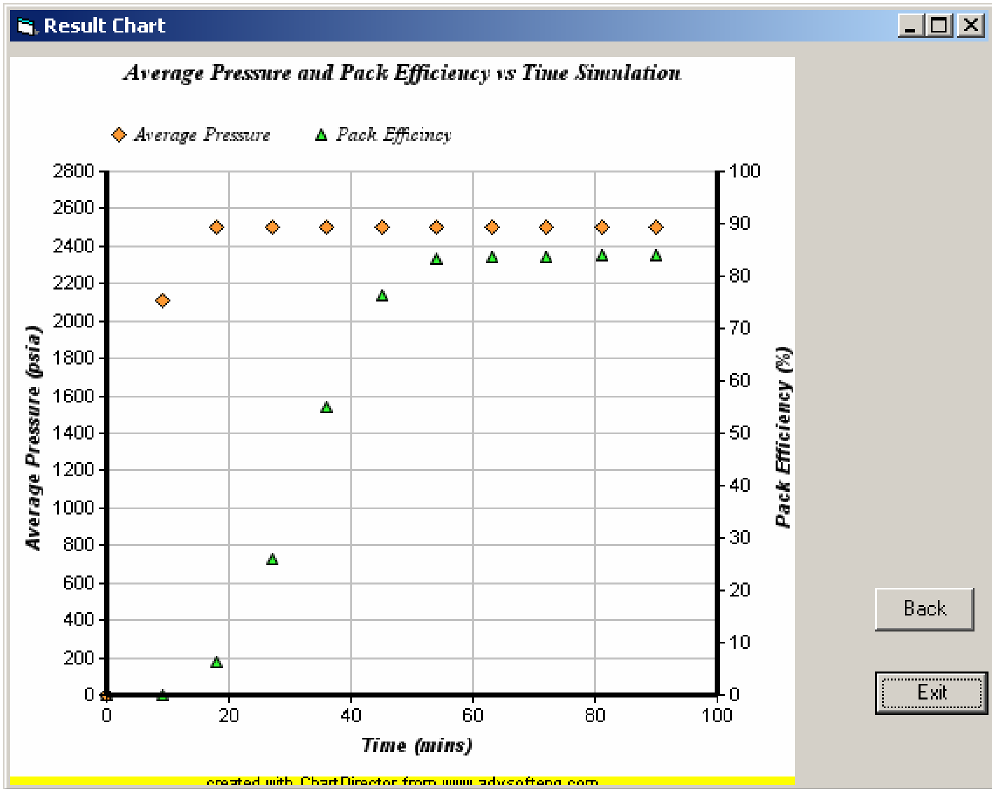


Figure 4.11: HGPSIM Final Simulation Result Screen

CHAPTER 5

MODEL VALIDATION AND SENSITIVITY STUDY

5.1 INTRODUCTION

The primary use of a gravel pack simulator is to be able to determine the main objectives of a gravel pack design. Conventional methods exist in the industry for determining different sections of the gravel pack design. The simulator will serve the purpose of confirming that these individually determined criteria will actually combine to produce an effective gravel pack.

The main objectives of a gravel pack design include

- Gravel pack method selection: this entails selecting between high rate water pack (HRWP) characterized by high flow rate and low gravel concentration between 1 to 2 ppg, Slurry pack involving polymer based fluid and high gravel concentration between 8 to 15 ppg and Frac-Pack involving pumping at fracturing rates with crosslinked polymer gel, high gravel concentrations.

- Gravel selection: involves selection of size and quality of gravel. The gravel size should be such that permeability of pack is retained and formation sand is prevented from passing through the gravel pack. Gravel size is usually selected based on sieve analysis on formation sand samples from conventional, sidewall cores or produced sand. Available methods used include Coberly method which uses largest sand grains to size gravel and the Saucier method which uses a pack-to-formation grain size ratio of 5 to 6 based on the median grain size of both pack and formation.

- Gravel Pack Screen Selection: This includes screen length, screen outer diameter, slot size and screen type. The slot size is selected based on a rule of thumb of ½ to 2/3 of smallest gravel grain. The available screen types include standard, prepacked, and wire mesh screens. The outer diameter must be such that enough clearance exists for gravel transportation and deposition.

- Carrier Fluid Selection: The carrier fluid must have enough viscosity to transport and deposit gravel at the injection rate.

This chapter illustrates how the simulator can be used to ascertain the above stated objectives. The first part of the chapter presents the validation of the developed simulation model by simulating three actual horizontal well gravel pack jobs from literature. The results demonstrate the validity of the solution routine and the capability of the HGPSIM program for simulating the gravel transportation and deposition during the gravel pack process and determining critical job parameters.

The datasets used in validating the model are presented in the examples 1, 2 and 3. The first example [31] is a 6-5/8 in 1778-ft horizontal section that was completed open hole in deepwater environment. The relevant input data for this example into the simulator is presented in Table 5.1. The second example [31] is an 8.5-in 2266-ft horizontal section and completed open hole. The input data for this example is presented in Table 5.2. Finally, the third example [34] is a 8.5-in 1073-ft open hole completion with input data shown in Table 5.3. Appendix A contains the figures obtained from the HGPSIM for examples 1 through3.

The second part of the chapter presents the result of sensitivity study conducted using the HGPSIM simulator. Job design parameter values known to affect the result of gravel pack operations including slurry density, gravel concentration and injection rate were included in the study. This exercise is very useful in designing gravel pack jobs and confirming that the individually obtained design parameters will actually produce an effective gravel pack.

5.2 EXAMPLE 1

The first example is a 6-5/8-in, 1778-ft horizontal section that was completed open hole in a deepwater environment. The well completed offshore has a reservoir depth of 6939-ft TVD. The total length of the horizontal section is 1778-ft. Table 5.1 shows the relevant data used for simulating the gravel pack job.

This example was simulated first with the settling effect turned on in the simulator and then with the settling effect turned off. Figures A-5.1 through A-5.20 show snapshots of the simulator at a different time steps during the simulation and some simulation results in form of charts. These show the progress of the gravel transportation and deposition in the horizontal wellbore with time for both simulation runs completed for this example.

Figures A-5.1 through A-5.6 show the simulation progress for the case that considered settling effect in the calculations. The simulator computed a final pack efficiency of 71.52% with a settling factor of 22.76% for this case. Figures A-5.7 through A-5.9 show simulation result charts, displaying the plot of pack efficiency and average cell pressure with axial distance (wellbore length) for different time step. Figure A-5.10 shows the final simulation result plot of average surface injection pressure and pack efficiency vs. time.

Figures A-5.11 through A-5.16 show the simulation progress for the case that neglected settling effect in the calculations. The simulator computed a final pack efficiency of 74.6% for this case. Figures A-5.17 through A-5.19 show some of the simulation results charts displaying the plot of pack efficiency and average cell pressure with wellbore length. Figure A-5.20 shows the final simulation result plot of average surface injection pressure and pack efficiency versus time.

Table 5.1: Input Data for Example 1[31]

Parameter	Value	Unit
Reservoir Pressure	4200	psia
Formation Depth	7500	ft
Vertical Section TVD	6939	ft
Vertical Section MD	7500	ft
Horizontal Section Open Hole Diameter	6 5/8	in
Screen Diameter	5.16	in
Screen Length	200	ft
Tell Tail Diameter	4.25	in
Tell Tail Length	200	ft
Horizontal Section Length	1778	ft
Carrier Fluid Density	9.1	ppg
Carrier Fluid Viscosity	1	cp
Gravel Density	22.08	ppg
Gravel Average Diameter	0.025	in
Slurry Density	11	ppg
Gravel Concentration	0.5	ppg
Slurry Injection Rate	2	bpm
Surface Injection Pressure	500	psia

5.3 EXAMPLE 2

The second example is an 8.5-in, 2266-ft horizontal section that was completed open hole in a deepwater environment. The well completed offshore has a reservoir depth of 6939-ft TVD. The total length of the horizontal section is 2266-ft. Table 5.2 shows the relevant data used for simulating the gravel pack job.

This example was also simulated first with the settling effect turned on in the simulator and then with the settling effect turned off. Figures A-5.21 through A-5.40 showed animated snapshots of the simulator at a different time step during the simulation and the charts of some result. These figures present the progress of the gravel transportation and deposition in the horizontal wellbore with time for both simulation runs completed for this example.

Figures A-5.21 through A-5.26 showed the simulation progress for the case that considered settling effect in the calculations. The simulator computed a final pack efficiency of 61.25% with a settling factor of 13.79% for this case. Figures A-5.27 through A-5.29 showed some of the simulation result charts displaying the plot of pack efficiency and average cell pressure with axial distance (wellbore length). Figure A-5.30 shows the final simulation result plot of average surface injection pressure and pack efficiency vs. time.

Figures A-5.31 through A-5.36 show the simulation progress for the case that neglected settling effect in the calculations. The simulator computed a final pack efficiency of 61.43% for this case. Figures A-5.37 through A-5.39 show some of the simulation results charts displaying the plot of pack efficiency and average cell pressure with wellbore length. Figure A-5.40 shows the final simulation result plot of average surface injection pressure and pack efficiency vs. time.

Table 5.2: Input Data for Example 2 [31]

Parameter	Value	Unit
Reservoir Pressure	4200	psia
Formation Depth	7500	ft
Vertical Section TVD	6939	ft
Vertical Section MD	7500	ft
Horizontal Section Open Hole Diameter	8 1/2	in
Screen Diameter	6.15	in
Screen Length	120	ft
Tell Tail Diameter	5	in
Tell Tail Length	120	ft
Horizontal Section Length	2266	ft
Carrier Fluid Density	8.8	ppg
Carrier Fluid Viscosity	1	cp
Gravel Density	22.08	ppg
Gravel Average Diameter	0.025	in
Slurry Density	9.38	ppg
Gravel Concentration	1	ppg
Slurry Injection Rate	6.45	bpm
Surface Injection Pressure	700	psia

5.4 EXAMPLE 3

The last example is a 8.5-in 1073-ft horizontal section that was completed open hole in a deepwater environment. The well completed offshore has a reservoir depth of 5030-ft TVD. The total length of the horizontal section is 1073-ft. Table 5.3 shows the relevant data used for simulating the gravel pack job.

This example was simulated first with the settling effect turned on in the simulator and then with the settling effect turned off. Figure A-5.41 through A-5.60 show snapshots of the simulator at different time steps during the simulation. The figures show the progress of the gravel transportation and deposition in the horizontal wellbore with time for both runs completed for this example.

Figures A-5.41 through A-5.46 show the simulation progress for the case that considered settling effect in the calculations. The simulator computed a final pack efficiency of 99.38% with a settling factor of 44.98% for this case. Figures A-5.47 through A-5.49 show some of the simulation results charts displaying the plot of pack efficiency and average cell pressure with axial distance. Figure A-5.50

shows the final simulation result plot of average surface injection pressure and pack efficiency vs. time.

Figures A-5.51 through A-5.56 show the simulation progress for the case that neglected settling effect in the calculations. The simulator computed a final pack efficiency of 99.42% for this case. This is close to the packing efficiency of 105% obtained in the reference. Figure A-5.57 though figure A-5.59 show some of the simulation result charts displaying the plot of pack efficiency and average cell pressure with wellbore length. Figure A-5.60 shows the final simulation result plot of average surface injection pressure and pack efficiency vs. time.

Table 5.3: Input Data for Example 3 [34]

Parameter	Value	Unit
Reservoir Pressure	2773	psia
Formation Depth	7847	ft
Vertical Section TVD	5030	ft
Vertical Section MD	7847	ft
Horizontal Section Open Hole Diameter	8.5	in
Screen Diameter	4.25	in
Screen Length	100	ft
Tell Tail Diameter	4.0	in
Tell Tail Length	100	ft
Horizontal Section Length	1073	ft
Carrier Fluid Density	9.1	ppg
Carrier Fluid Viscosity	1	cp
Gravel Density	22.08	ppg
Gravel Average Diameter	0.025	in
Slurry Density	11	ppg
Gravel Concentration	0.5	ppg
Slurry Injection Rate	4.3	bpm
Surface Injection Pressure	1000	psia

5.5 SENSITIVITY STUDY

A sensitivity study was conducted using the HGPSIM simulator that was developed in this project. The base data used for this study is presented in Table 5.3. The study considered the effect of perturbation factor, gravel concentration, viscosity and injection rate on the pack efficiency obtained from the simulator. The following sections present the result of each variable studied.

5.5.1 EFFECT OF PERTURBATION FACTOR ON MODEL RESULT

In estimating the partial derivatives in Eq. 3.22 of section 3.5, a small perturbation is assumed for each of the dependent variables. The numerical solution to the modeling equations is expected to be independent of the amount of perturbation used in calculating the partial derivatives.

To study this effect, a sensitivity study was conducted on the perturbation factor to evaluate it's effect on the pack efficiency. Perturbation factors of 10, 20 and 30% were used for 30 simulation runs with varying input parameters. Fig 5.1 shows the results of the simulation runs. The figure indicates that the perturbation factor have almost no effect on the result from the simulator. The curves for the 10, 20 and 30% perturbation factor almost match for both cases, with or without settling effect.

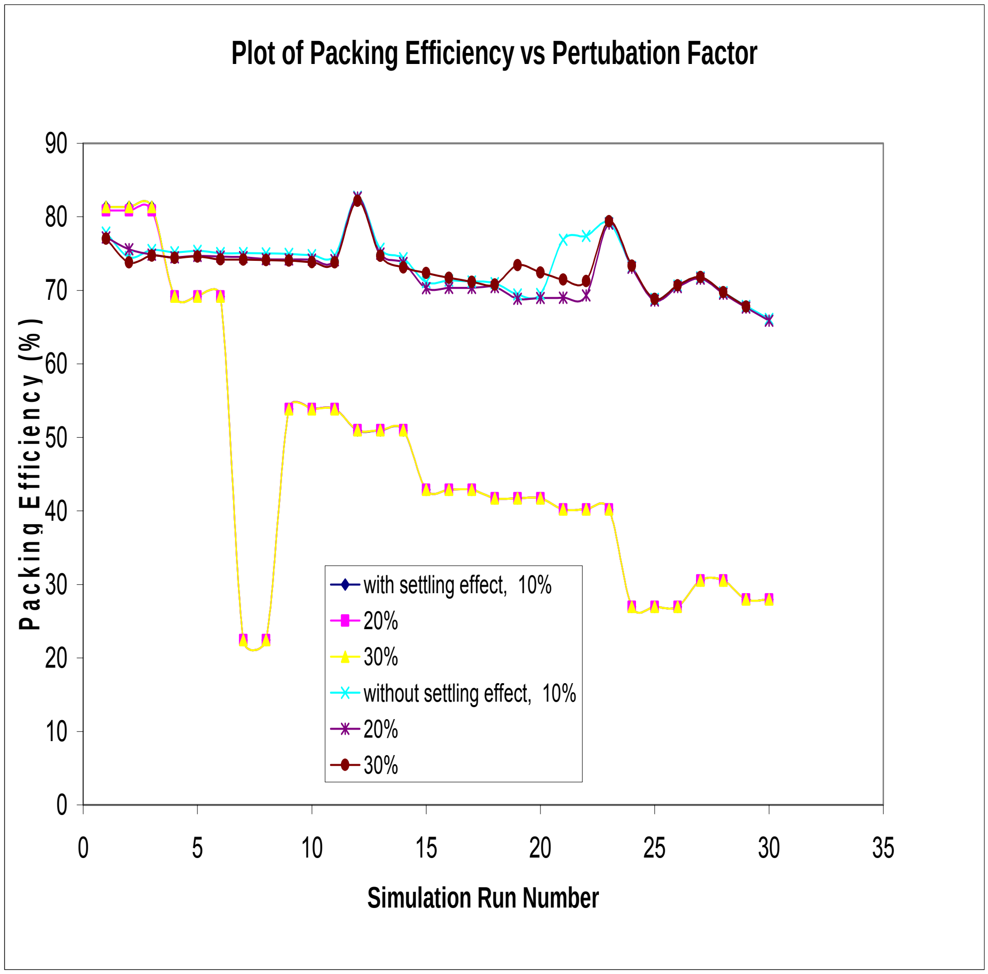


Figure 5.1: Effect of Perturbation Factor on model result

5.5.2 EFFECT OF GRAVEL CONCENTRATION AND INJECTION RATE ON SETTLING FACTOR

The effect of injection rate and gravel concentration on settling factor estimated by simulator was investigated. The result is presented in Fig. 5.2. The injection rate was varied between 1 and 8 bpm while gravel concentration was varied between 0.5 and 10 ppg. The figure shows that in general, settling factor decreases with increasing gravel concentration and injection rate.

At low concentrations of between 0.5 to 3.5 ppg, the lower the injection rate, the higher the predicted settling factor. As injection rate increases, there is an almost constant steady decline in the settling factor. At higher gravel concentration above 3.5 ppg, there is a steady decline in the settling factor with increasing injection rate until a plateau is reached where very sudden decline in the settling factor is observed. This sudden decline continues until a second plateau is reached where a steady decline in the settling factor resumes.

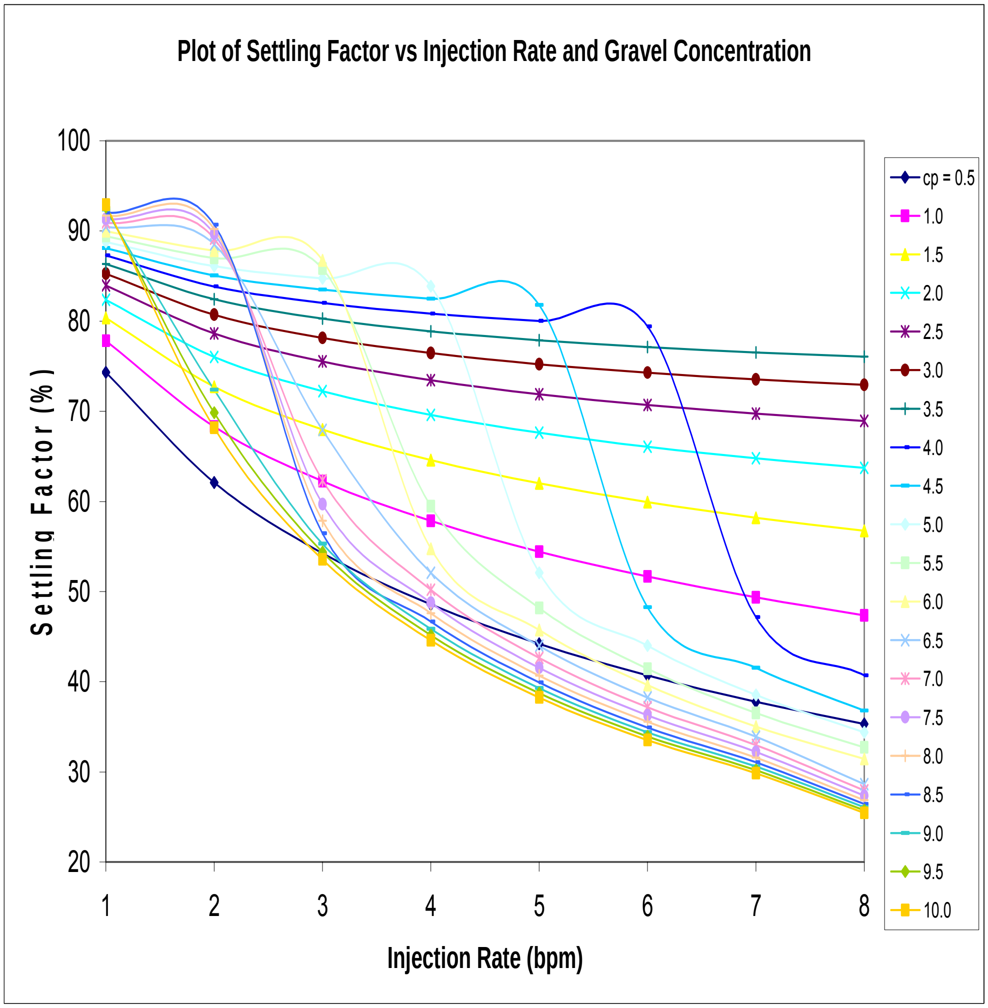


Figure 5.2: Effect of Injection rate and Gravel Concentration on Settling Factor

5.5.3 EFFECT OF GRAVEL CONCENTRATION AND INJECTION RATE ON PACK EFFICIENCY

The effect of injection rate and gravel concentration on simulator estimated pack efficiency was studied by varying injection rate between 1 and 8 bpm while varying gravel concentration between 0.5 and 10 ppg. Two sets of simulation runs were done, with and without settling effect. Fig. 5.3 shows the results of the runs without settling effect while Fig. 5.4 shows those with settling effect.

Fig. 5.3 shows that at high injection rates of about 4 to 8 bpm, the model predicted pack efficiencies were very close for the entire range of gravel concentrations considered. At lower injection rate than 4 bpm, there are slight variations in the predicted pack efficiencies with the highest values predicted for the lowest gravel concentration. This is in agreement with the general acceptable practice for gravel packing horizontal wells and earlier experimental observations in the literature.[3,5,10,11,13,33.]

With settling effect under consideration, the result is very different. Fig. 5.4 shows that the model predicted pack efficiency for these cases are consistently higher than the corresponding cases without settling effect throughout the range of injection rates and gravel concentration considered. This implies that available horizontal well gravel pack models that neglects settling effect consistently under predicts the pack efficiency obtained from gravel packing.

Fig. 5.4 shows that at low injection rates between 1 to 2 bpm, low gravel concentrations predicted slightly higher pack efficiency. As injection rate increases, the pack efficiency obtained from higher gravel concentration increases steadily while that from low concentrations decreases. The declines are however, oscillatory in nature.

This suggests that higher pack efficiency can be obtained at high injection rate and high gravel concentration. While these operating conditions are desirable in terms of good packing efficiency and lower job cost from lower pump time, such conditions are practically impossible because of the high friction pressure involved with pumping at such high concentration and high injection rate. High friction pressure is directly related to exorbitant pumping cost and possible failure of equipments in the wellbore. For these reasons, conditions are usually maintained at reasonably low injection rates and gravel concentrations in selecting the optimum operating conditions.

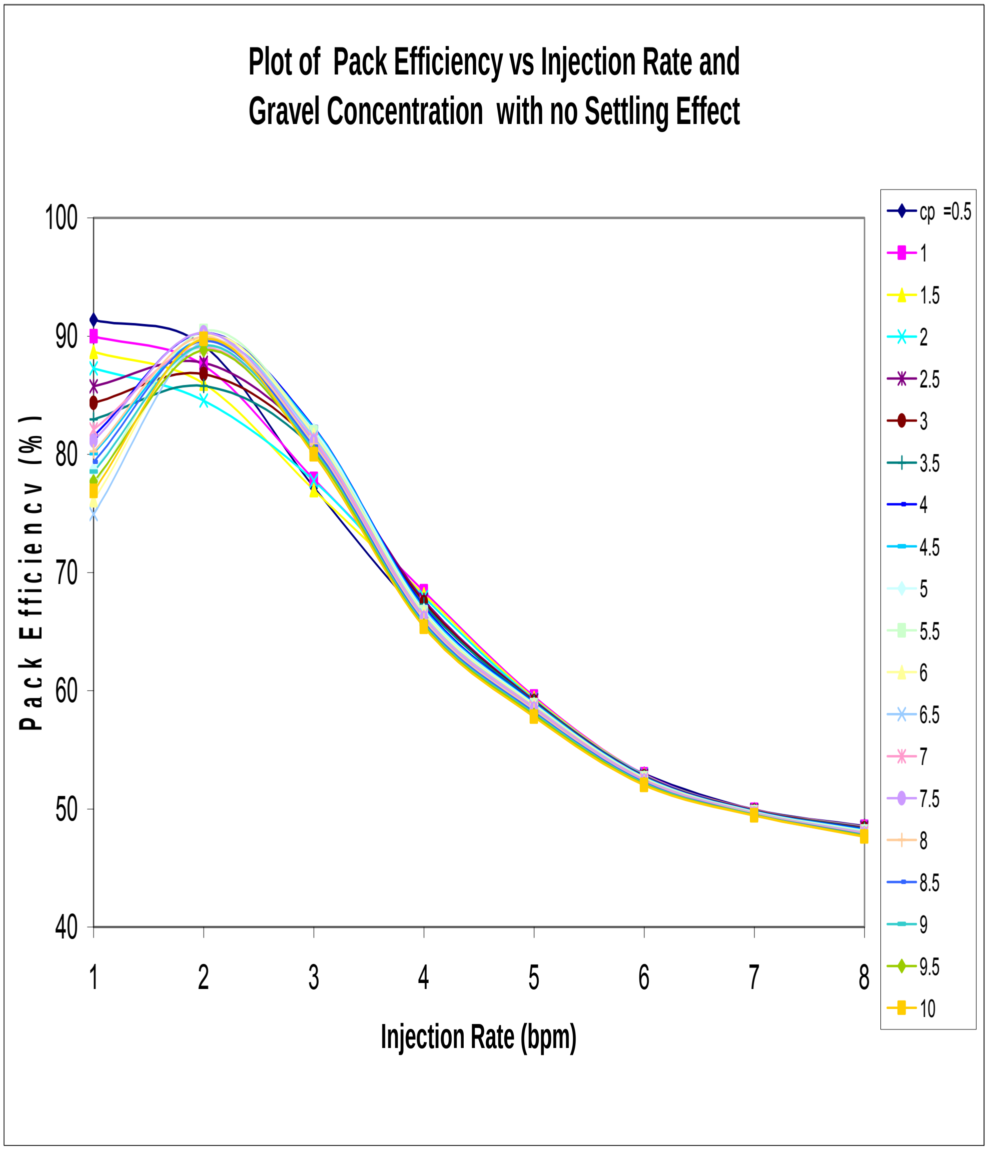


Figure 5.3: Effect of Gravel Concentration and Injection rate on Pack Efficiency with no settling effect

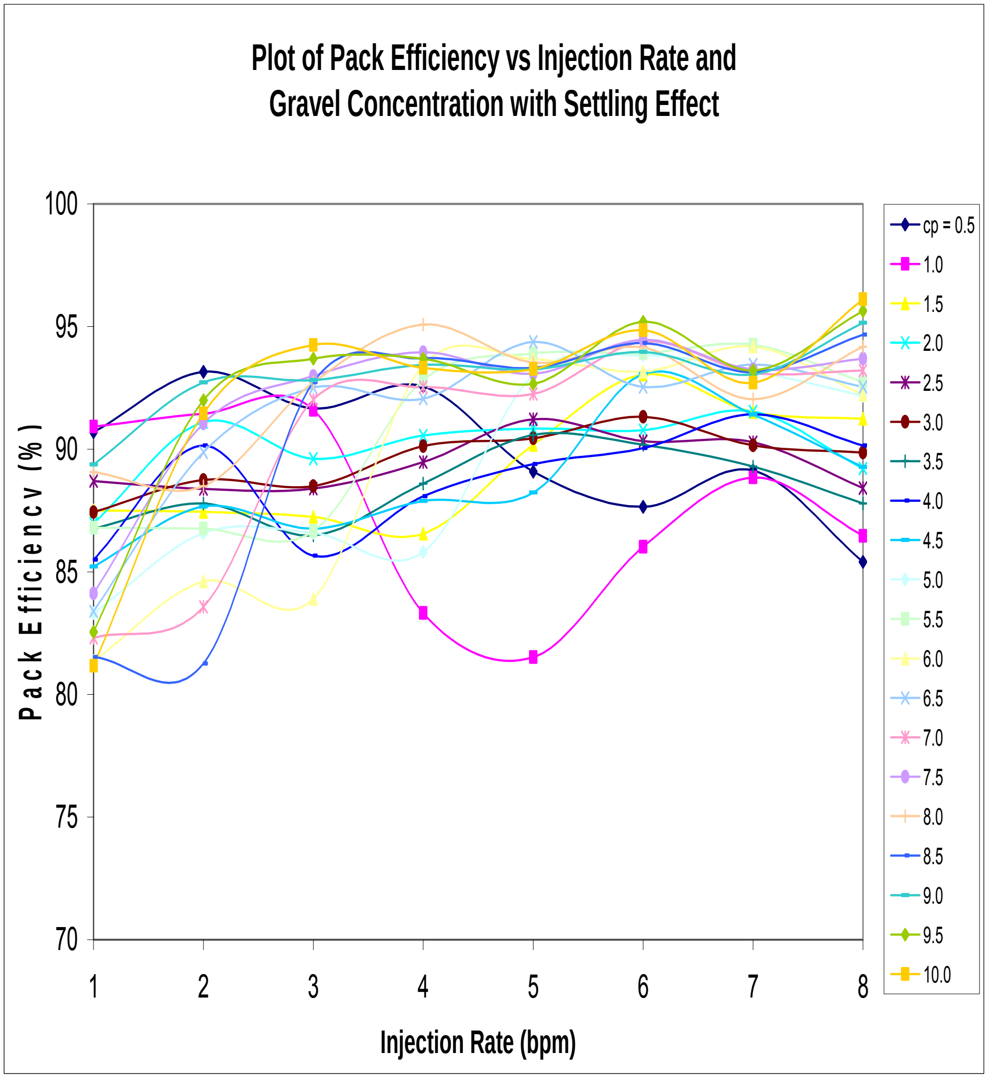


Figure 5.4: Effect of Gravel Concentration and Injection rate on Pack Efficiency with settling effect

5.5.4 EFFECT OF VISCOSITY ON PACK EFFICIENCY

The effect of carrier fluid viscosity on pack efficiency was studied by varying the viscosity between 1 and 51 cp, injection rate between 1 and 4 bpm and gravel concentration between 0.5 and 4.0 ppg. The result of the simulation runs are presented in Figs. 5.5 and 5.6. These figures indicate that for the wide range of variables considered, viscosity has little effect on the pack efficiency obtained for both cases, with and without settling effect. The pack efficiency in both cases stayed above 80% over the entire range of input parameter considered. The low viscosity fluids at low gravel concentration have higher packing efficiencies which are more stable over the entire range of injection rate. This supports the usual practice in horizontal well gravel pack in which low gravel concentrations are used with the carrier fluid of mostly water or brine.[31, 34]

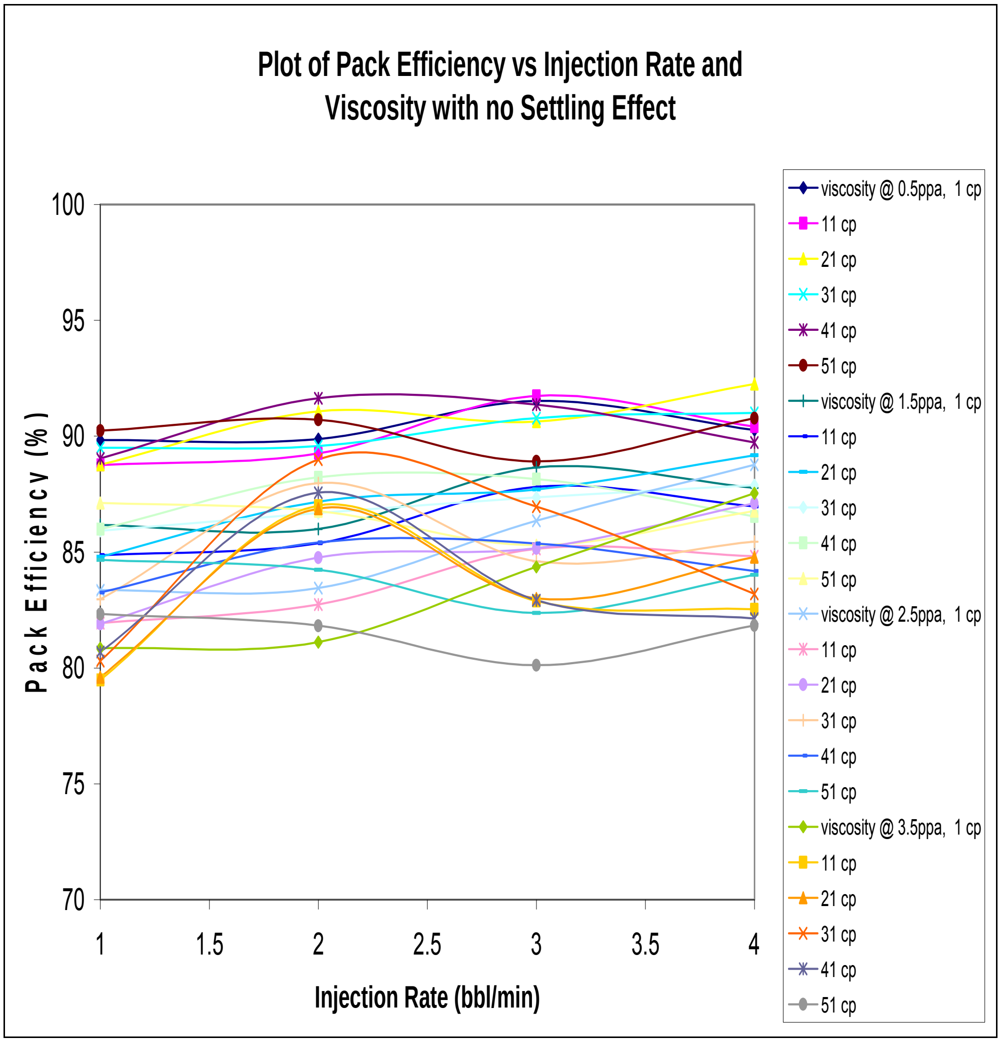


Figure 5.5: Effect of Viscosity on Pack Efficiency with no Settling

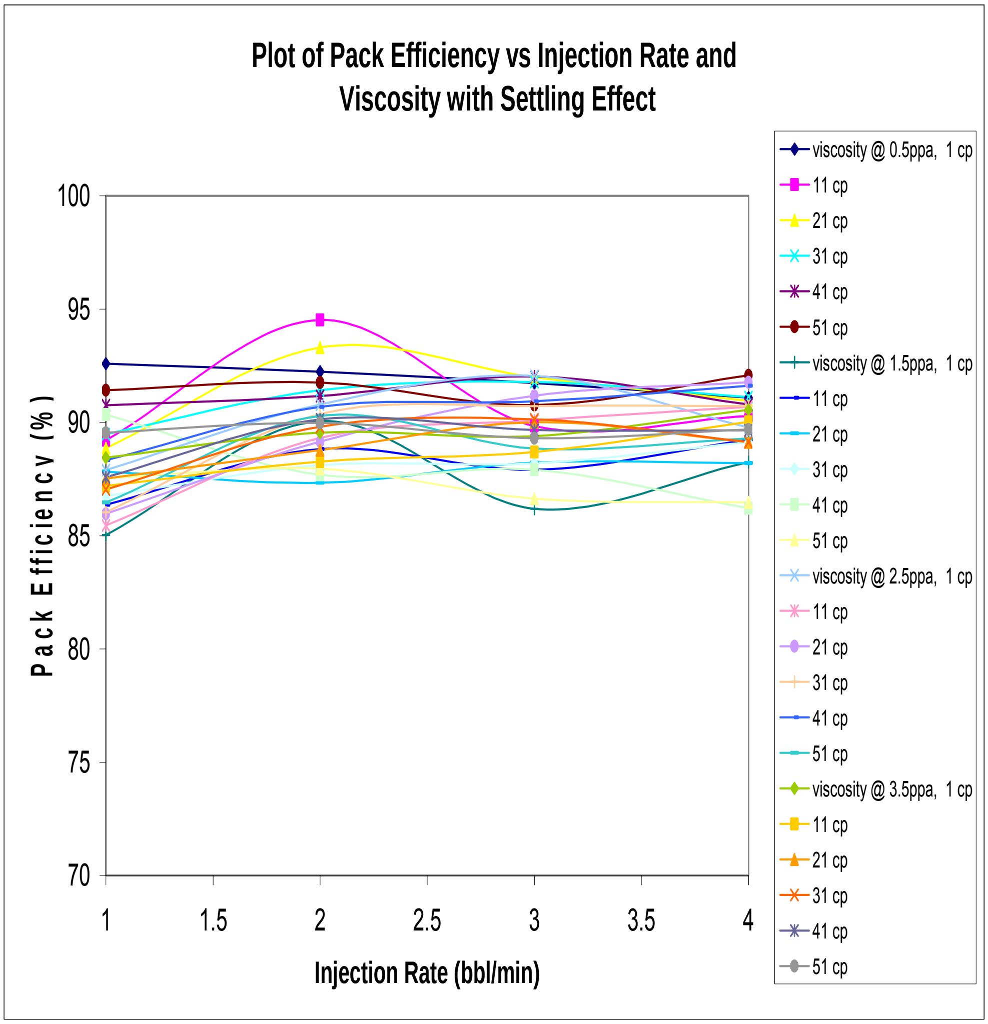


Figure 5.6: Effect of Viscosity on Pack Efficiency with Settling

5.6 SUMMARY

This chapter demonstrates the application of the developed simulator, HGPSIM, for simulating the horizontal well gravel pack process. The model validation cases show the type of job performance data that is obtainable from the simulator. The gravel pack efficiency and system pressure at any point in time and distance was determined and monitored as a function of the various job parameters.

Comparison was be made between results obtained with and without considering the settling effects in the vertical section of the wellbore. The results indicated that these effects vary depending on other variables in the system, including the wellbore geometry, formation properties and gravel pack job parameters thereby supporting the idea that settling effects has to be considered to determine the effect on the gravel pack result.

An animated depiction of the wellbore status at various time steps, showing the progress of the gravel pack, was developed as part of the HGPSIM program, and is a very useful tool in the gravel pack design process. This tool gives a visual depiction of the instantaneous results of the simulation runs at different time steps. It gives the Engineer a visual picture of the wellbore at different time step, thus allowing him or her, the ability to determine where and when a premature bridge is likely to occur within the system.

Finally, a sensitivity study, similar to what will be required in designing a gravel pack job was conducted. This study shows that several parameters have significant effect on the result of the gravel pack job and needs to be considered in designing one.

CHAPTER 6

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

6.1 SUMMARY

A 3-D numerical simulation model for horizontal well gravel pack is developed for simulating the gravel pack process and determining the effectiveness and pack efficiency of the gravel pack job. This model, being a true 3-D model can be used to simulate special situations in which significant leakage into the formation occur during gravel packing.

The simulation model is particularly suitable for deep horizontal wells because the hindered settling effect prevalent in the long vertical section of deep horizontal wells is accounted for in the modeling equations.

The model was developed based on the fundamental laws of physics by solving for an overall conservation of forces during the placement of the gravel in the horizontal well bore. The equations of continuity, motion, and deposition are simultaneously solved to give a 3-D solution of the gravel pack process. The settling effect of slurry using the type of fluid, proppant and flow conditions existing in gravel pack operations in horizontal wells were also included in the model.

The modeling equations consist of five partial differential equations with five dependent variables, all of which have to be solved simultaneously. The system of partial differential equations was approximated using the central difference finite difference approximations method. The resulting system of non-linear algebraic equations was then solved using the Newton's method of solving non-linear simultaneous equations. A solution algorithm was developed and implemented using a computer program.

HGPSIM, a horizontal well gravel pack simulator was developed to implement the solution algorithm. The usefulness and validity of the model was also illustrated by simulating some horizontal gravel pack jobs reported in the literature using the developed simulator.

6.2 CONCLUSIONS

1. A 3-D numerical simulation model is developed that offers a significant improvement over the currently available empirical, 1-D, 2-D and pseudo 3-D horizontal well gravel pack models.

2. The developed horizontal well gravel pack model can be used for simulating the gravel pack process and determining the effectiveness and total pack efficiency of the gravel pack job. The model predicted the change in pack efficiency with wellbore length, giving a measure of the gravel pack height achieved along the horizontal wellbore in real time as gravel placement progresses.

3. The average pressure in each grid block is predicted allowing the possible prediction of high pressure points and possible blockage in the wellbore during the gravel pack process.

4. The sensitivity study conducted showed that available horizontal well gravel pack models that neglects settling effect consistently under predicts the pack efficiency. The

settling factor decreases with increasing gravel concentration and injection rate.

5. Without considering settling effect at high injection rates, the model predicted pack efficiencies were very close for the entire range of gravel concentrations considered. At lower injection rates, there are slight variations in the predicted pack efficiencies with the highest values predicted for the lowest gravel concentration. With settling effect under consideration, the model predicted pack efficiency for these cases are consistently higher than the corresponding cases without settling effect throughout the range of parameters considered.

6. Carrier Fluid viscosity has little effect on the pack efficiency obtained for both cases, with and without settling effect. The pack efficiency in both cases stayed above 80% over the entire range of input parameter considered.

6.3 RECOMMENDATIONS

1. The 3-D numerical simulation model developed in this study should be validated using a laboratory horizontal well model.

2. The model can be extended to account for non-Newtonian fluids that can be used in gravel pack jobs and to predict amount of fluid lost to the formation and thus estimate amount of damage resulting from fluid loss to the formation.

NOMENCLATURE

Alphabetical Symbols

A	= Area
C	= Concentration
C_p	= Input volume concentration
D, d	= Diameter
D_s	= Diameter of screen
D_g	= Average gravel diameter
D_{tp}	= Daimeter of Tailpipe
$d\theta$	= Grid size in angular direction
dr	= Grid size in radial direction
dz	= Grid size in axial direction
F	= Friction factor
E_S	= In-situ volume concentration
g_c	= Gravitational constant
g_r	= Gravitational constant in r direction
g_z	= Gravitational constant in z direction
g_θ	= Gravitational constant in θ direction
H, h	= Height
k	= Permeability
K	= Consistency index
L, l	= Length
n	= Fluid constant xxx
Q, q	= Flow rate
p	= Pressure
P.E	= Packing Efficiency
R	= Open hole or casing radius
r	= Radial distance
r_C	= Open hole or casing radius
r_L	= Liner radius
r_S	= Screen radius
S.F	= Settling Factor
t	= Time
v_r	= Radial component of velocity
v_z	= Axial component of velocity
v_θ	= Angular component of velocity
v	= Velocity
v_s, v_t	= Single particle settling velocity
V	= Volume

Greek Symbols

θ	= Angular component
μ	= Viscosity
ρ	= Density
φ	= Available pore space for deposition

Subscripts

A	Annular
C	Casing
cf	carrier fluid
eff	Effective
lss	Fluid loss
O	initial condition
P	Particle
S	Slurry
Tp	Tailpipe
w	Wellbore

REFERENCES

1. Zaleski, Jr. ., T. E.,: "Sand Control Alternatives for Horizontal wells," Paper SPE 22053, SPE Technology Today Series 1991.
2. Fahd, R. A. and Brienen, J. W.,: "A comparison of deepwater sand control practices in the Gulf of Mexico," SPE 23772, presented at SPE Intl. Symposium on Formation Damage Control held in Lafayette, Louisiana, Feb. 26-27 1992.
3. Penberthy, W. L. , Bickham, K. L., Nguyen, H. T., and Paulley, T. A.,: "Gravel Placement in Horizontal Wells," SPE 31147, presented at SPE Intl. Symposium on Formation Damage Control held in Lafayette, Louisiana, 1996.
4. Mathis, S. P., Costa, L. A.G, Andrea N. de Sa, Tsuha, Hebert, S. A.,: "Horizontal Gravel Packing Successfully Moved to Deepwater Floating Rig Environment," SPE 56783, presented at the 1999 SPE Annual Technical Conference and Exhibition, Houston, TX, Oct 3-6.
5. Gruesbeck, C., Salathiel, W. M., and Echols, E. E.,: "Design of Gravel Packs in Deviated Wellbores," SPE 6805, Inst. Of Pet. Tech., Jan. 1979, pp 109 – 115.
6. Hodge, R. M.,: "Gravel Transport in Deviated Wellbores," SPE 10654 ,1982.

7. Elson, T. D., Darlington, R. H., and Mantooth, M. A.,: “High Angle Gravel Pack Completion Studies,” SPE 11012, presented at the 1982 SPE Annual Technical Conference and Exhibition, New Orleans, L.A. , September 26 - 29.
8. Skaggs, B. C.,: “Transport Efficiency of High Density Gravel Packing Slurries,” SPE 12480, SPE Intl. Symposium on Formation Damage Control held in Bakersfield, CA 1984.
9. Winterfeld, P. H., and Schroeder, D. E. Jr.,: “ Numerical Simulation of Gravel Packing,” SPE 19753, SPE Production Engineering, Aug., 1992.
10. Peden, J. M., Russell, J., Oyeneyin, M. B.,: “A Numerical Approach to the Design of a Gravel Pack for effective Sand Control in Deviated Wells,” SPE 13084, presented at the 1984 SPE Annual Technical Conference and Exhibition, Houston, TX, September 16-19.
11. Wahlmeier, M. A., and Andrews, P. W.,: “Mechanics of Gravel Placement and Packing: A Design and Evaluation Approach,” SPE 16181, SPE Production Engineering, Feb., 1988.
12. Shryock, S. G.,: “Gravel Packing Studies in a Full-Scale Deviated Model Well bore,” JPT (March 1983) 693-09.

13. Forrest, J. K.,: "Horizontal Gravel Packing Studies in a Full-Scale Model Well bore, " SPE 20681, 1990.

14. Penberthy, W. L. Jr and Echols, E. E.,: "Gravel Placement in Wells," JPT (June 1993).

15. Nguyen P. D., Fitzpatrick H. J., Woodbridge G. A. and Reidenbach V. G.,: "Analysis of Gravel Packing Using 3-D Numerical Simulation," SPE 23792 presented at the SPE Formation Damage Symposium, Lafayette, LA, Feb., 26-27, 1992.

16. Sanders M.W., et al.,: "Gravel Pack Designs of Highly-Deviated Wells With an Alternative Flow-Path Concept," paper SPE 73743 presented at the SPE Formation Damage Symposium, Lafayette, LA, Feb., 20-21, 2002.

17. Hannah, R. R., and Harrington, L. J.,: "Measurements of Dynamics Proppant Fall Rates in Fracturing Gels Using a Concentric Cylinder Tester," JPT (May 1981) 909 –13.

18. Novotny, E. J., : "Proppant Transport," SPE 6813 presented at the 1977 SPE Annual Technical Conference and Exhibition, Denver, Oct. 9 - 12.

19. Kirkby, L. L. and Rockefeller, H. A.,: "Proppant Settling Velocities in Non-Flowing Slurries," SPE 13906 presented at the 1985 SPE/DOE Low permeability Gas Reservoirs Symposium, Denver, May 19-21.

20. Dunand, A. and Soucemarianadin, A.,: “Concentrated Effects on the Settling Velocities of Proppant Slurries,” paper SPE 14259 presented at the 1985 SPE Annual Technical Conference and Exhibition, Las Vegas, Sept. 22-25.

21. Steinour, H. H.,: “Rate of Sedimentation: Suspension of Uniform-Size Angular Particles, “Ind. & Chem. (Sept. 1944) 36, No. 9, 840-47.

22. Shah, S. N.,: “Proppant-Settling Correlations for Non-Newtonian Fluids,” SPEPE (Nov. 1986) 446-48.

23. McMechan, D. E., and Shah, S. N.,: “Static Proppant-Settling Characteristics of Non-Newtonian Fracturing Fluids in a Large Scale Test Model,” SPEPE (Aug. 1991) 305-312.

24. Shah, S. N.,: “Rheological Characterization of Hydraulic Fracturing Slurries,” SPEPE (May 1993) 123-30.

25. Lichtner, : “Water Resources Research,” Vol. 28, No. 12, December 1992, pp. 3135-3155.

26. Civan F.,: “A Multi-Purpose Formation Damage Model,” paper SPE 31101 presented at the SPE Formation Damage Symposium, Lafayette, LA, Feb., 14-15, 1996.

27. Govier, G. W. and K. Aziz.: “The Flow of Complex Mixtures in Pipes, “ Van Nostrand Reinhold Company, New York (1972).

28. Newitt, D. M., Richardson, J. F., Abbot, M. and Turtle, R. B.,: "Hydraulic Conveying of Solids in Horizontal Pipes," Trans., ICE (1955) 33, 93.

29. Bird, R.B., Stewart, W.E., and Lightfoot, E.N.,: "Transport Phenomena," Wiley, New York (1960).

30. G.C. Curtis, and P.O. Wheatley,; "Applied Numerical Analysis," Pg 204., October 1998, Published by Addison-Wesley.

31. Grigsby T. and Vitthal S.,: "Openhole Gravel Packing – An Evolving Mainstay Deepwater Completion Method," SPE 77433, 2002.

32. Ojo K.P. and Osisanya, S.O.,: "A Review of Current Gravel Transportation /Placement Models and Problems in Deepwater Horizontal Gravel-Pack Modeling" SPE 67295, 2001 presented at the 2001 SPE Production and Operations Symposium, Oklahoma City, March 24-27.

33. Perry R. H. and Green D. W. : "Perry's Chemical Engineer's Handbook, " Seventh Edition, McGraw-Hill Companies Inc., New York (1997).

34. Dickerson R. C., Ojo-Aromokudu O., Bodurin A. A., Passant K., Roane T., Penno A., and Fitzpatrick H.,: "Horizontal Openhole Gravel Packing with Reactive Shale Present," paper SPE 84164

presented at the 2003 SPE Annual Technical Conference and Exhibition, Denver, Oct. 5-8.

APPENDIX A: HGPSIM MODEL VALIDATION FIGURES

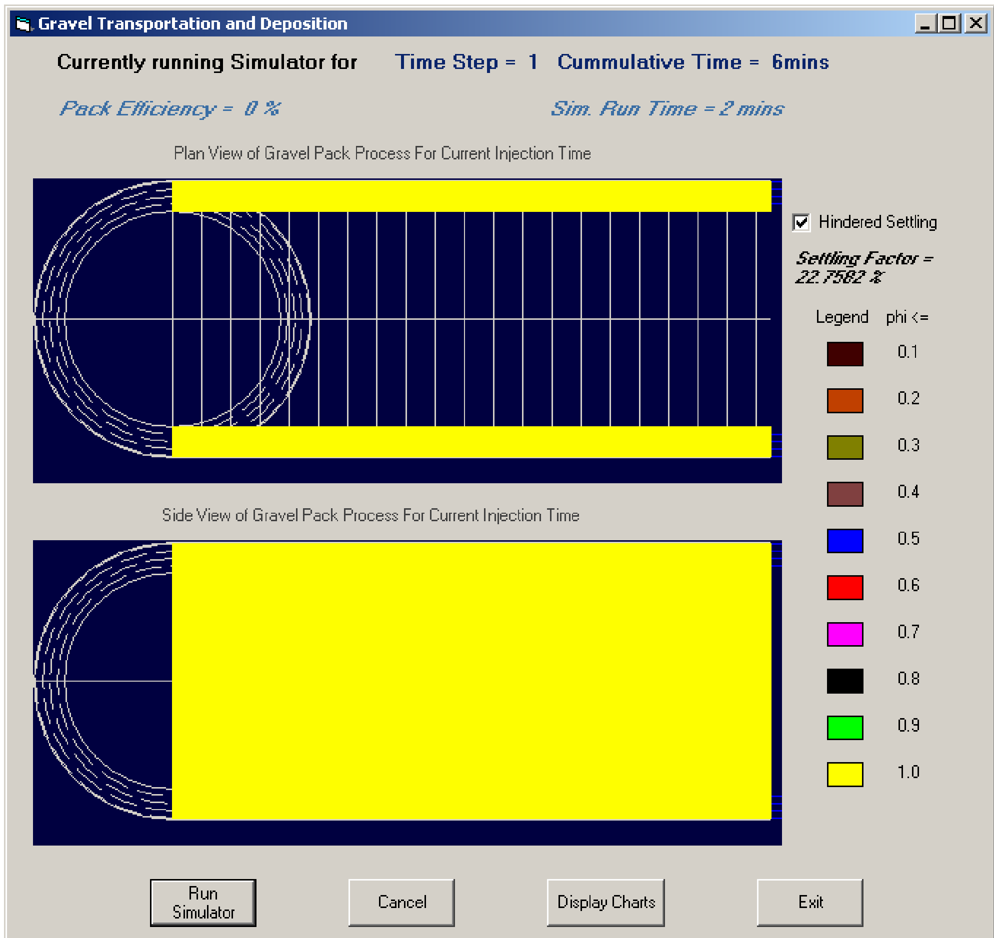


Figure A- 5.1: Example 1 Simulation in Progress for time step = 1

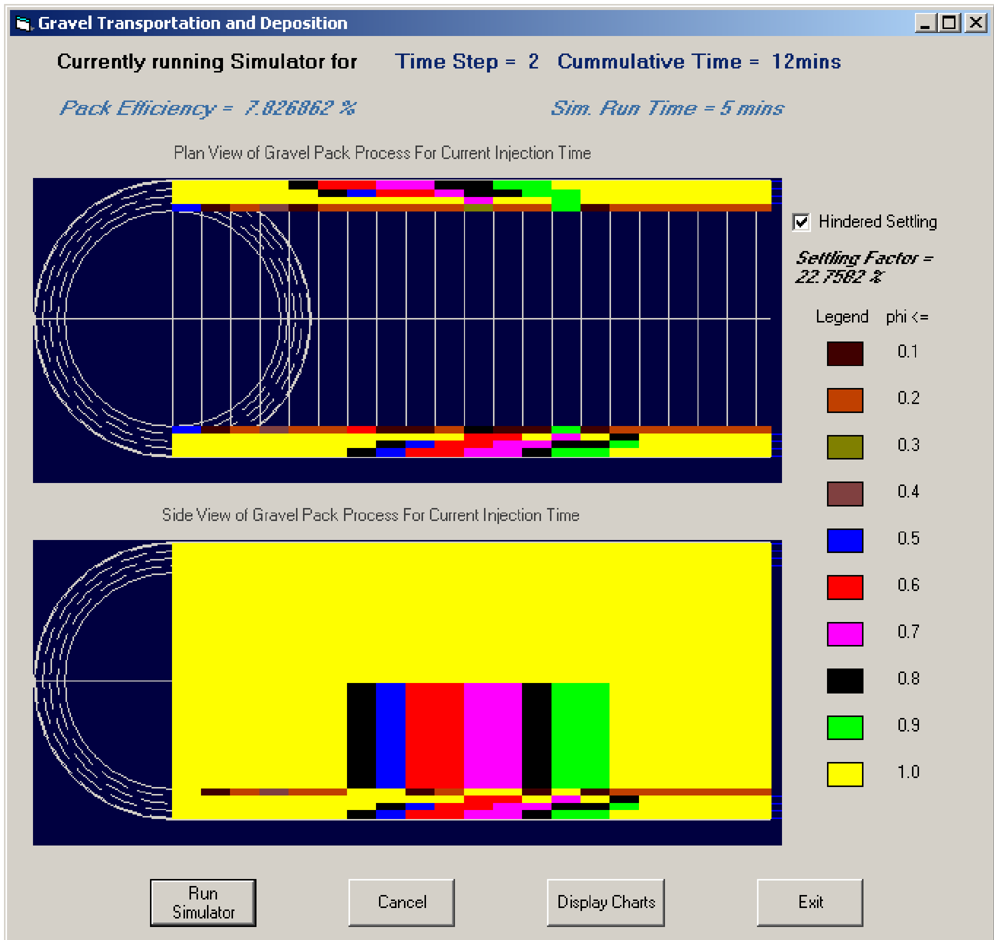


Figure A- A- 5.2: Example 1 Simulation in Progress for time step = 2

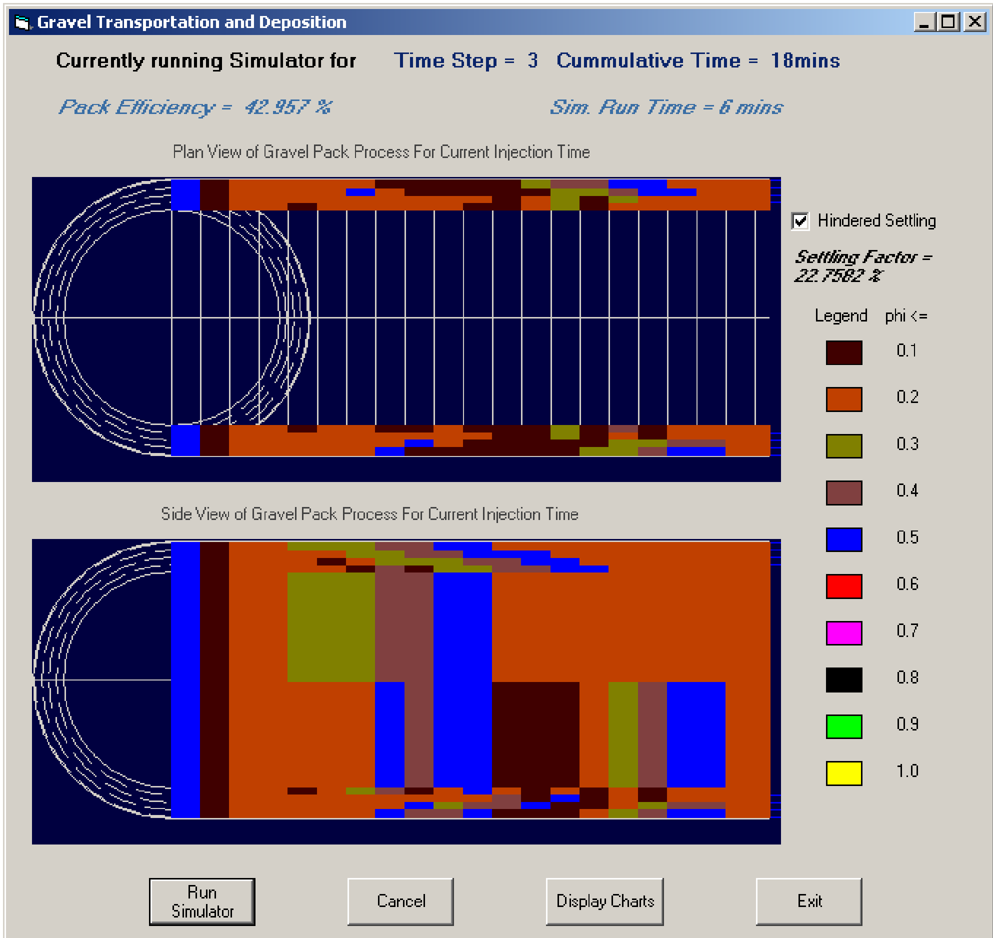


Figure A- 5.3: Example 1 Simulation in Progress for time step = 3

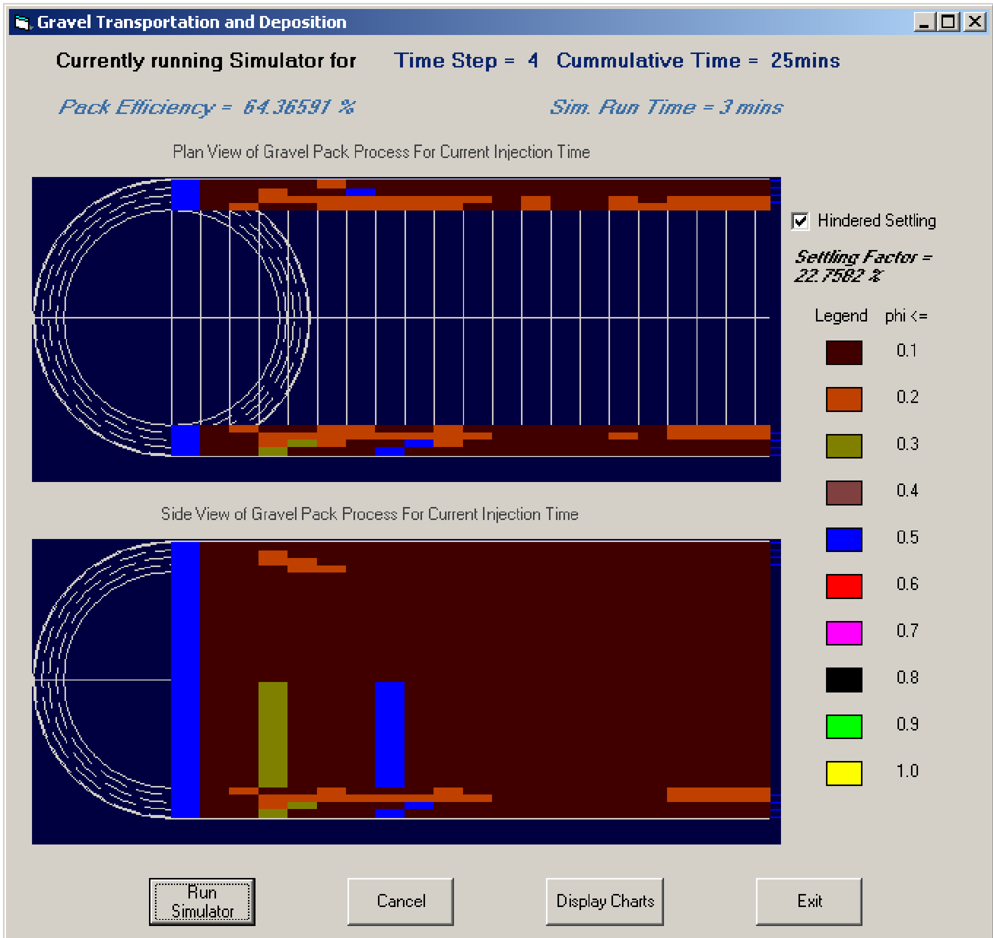


Figure A- 5.4: Example 1 Simulation in Progress for time step = 4

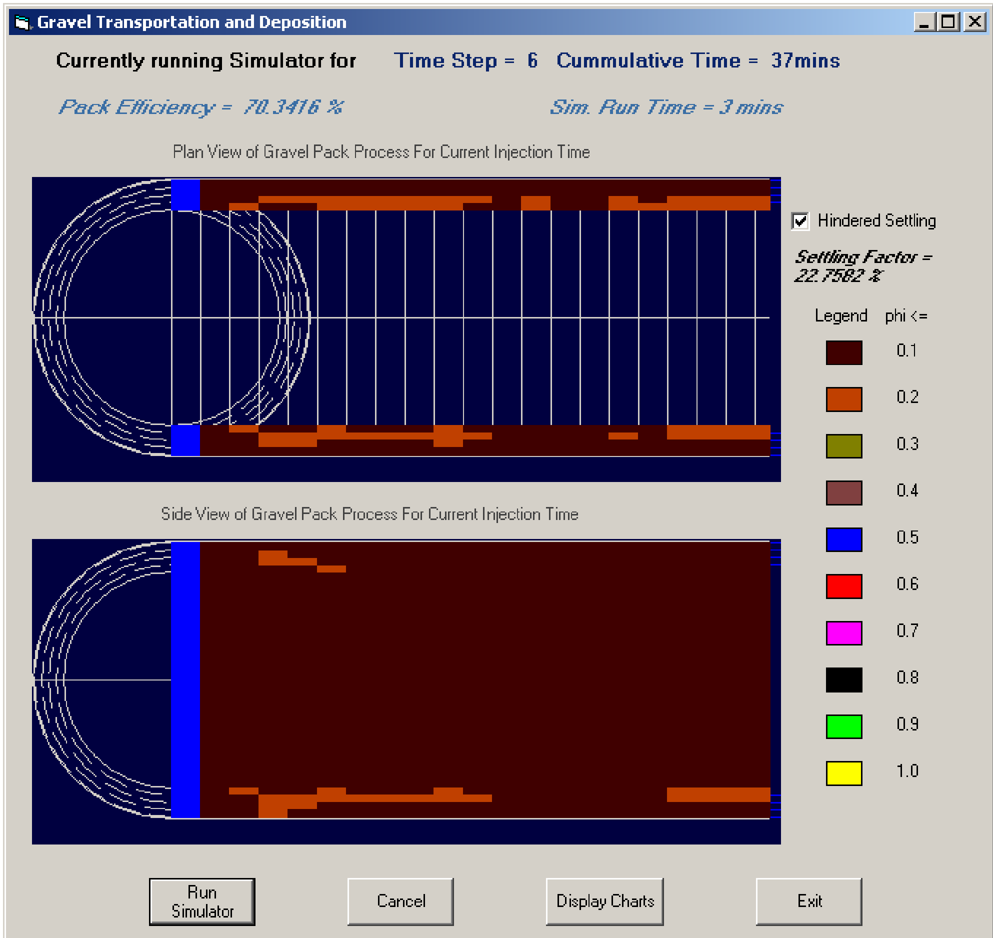


Figure A- 5.5: Example 1 Simulation in Progress for time step = 6

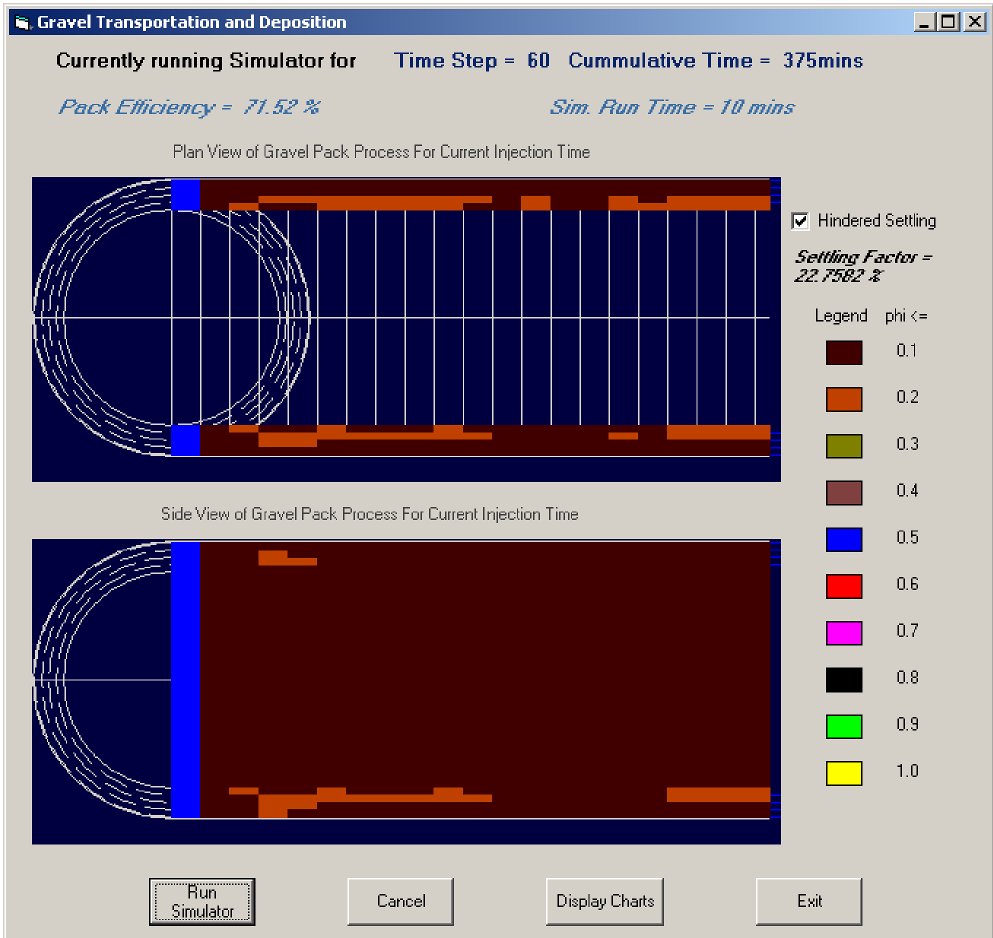


Figure A- 5.6: Example 1 Simulation in Progress for time step = 60

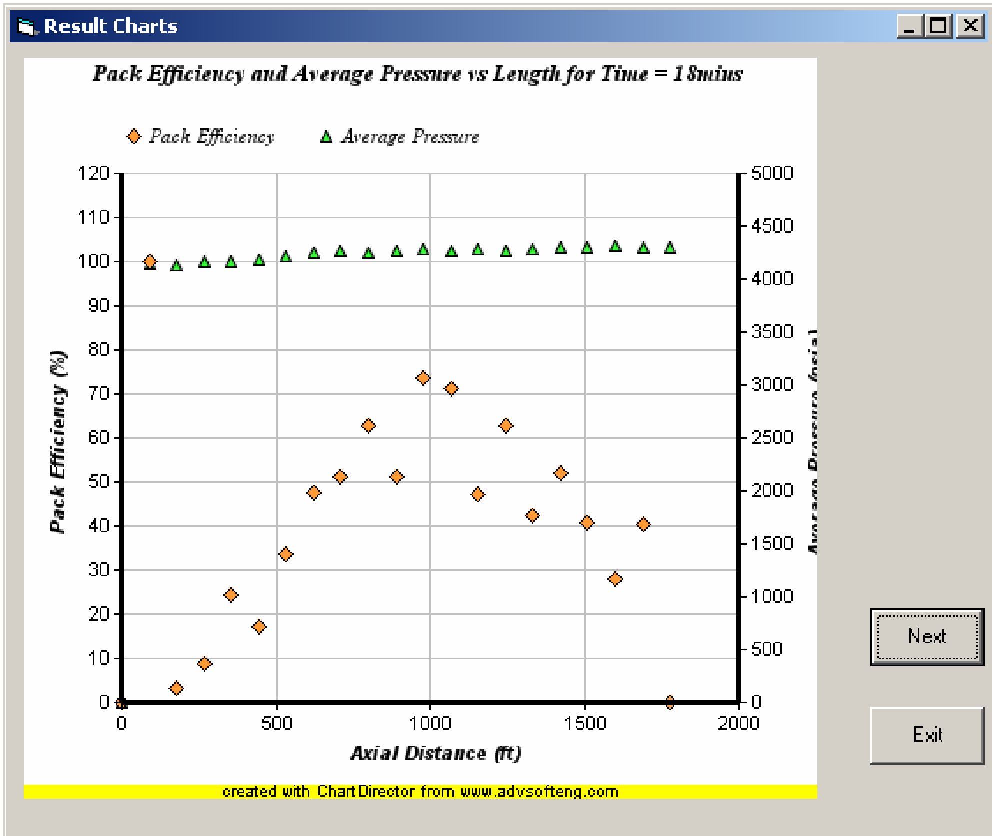


Figure A- 5.7: Example 1 Result Charts Screen at Time = 18min

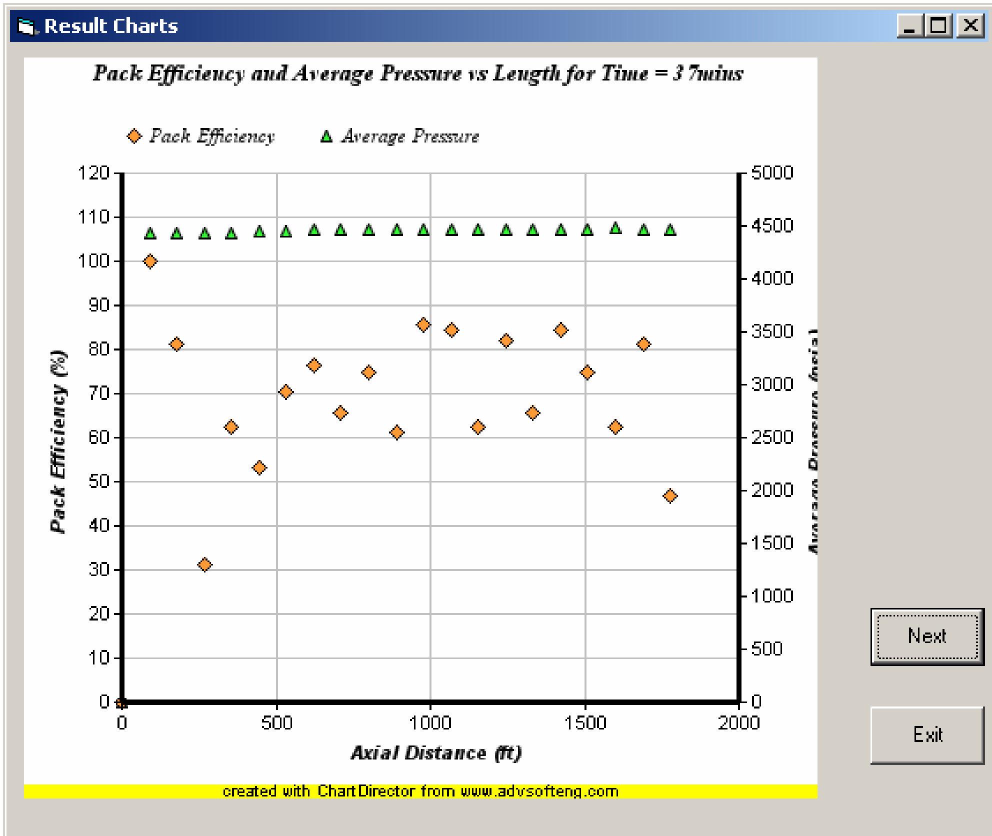


Figure A- 5.8: Example 1 Result Charts Screen at Time = 37min

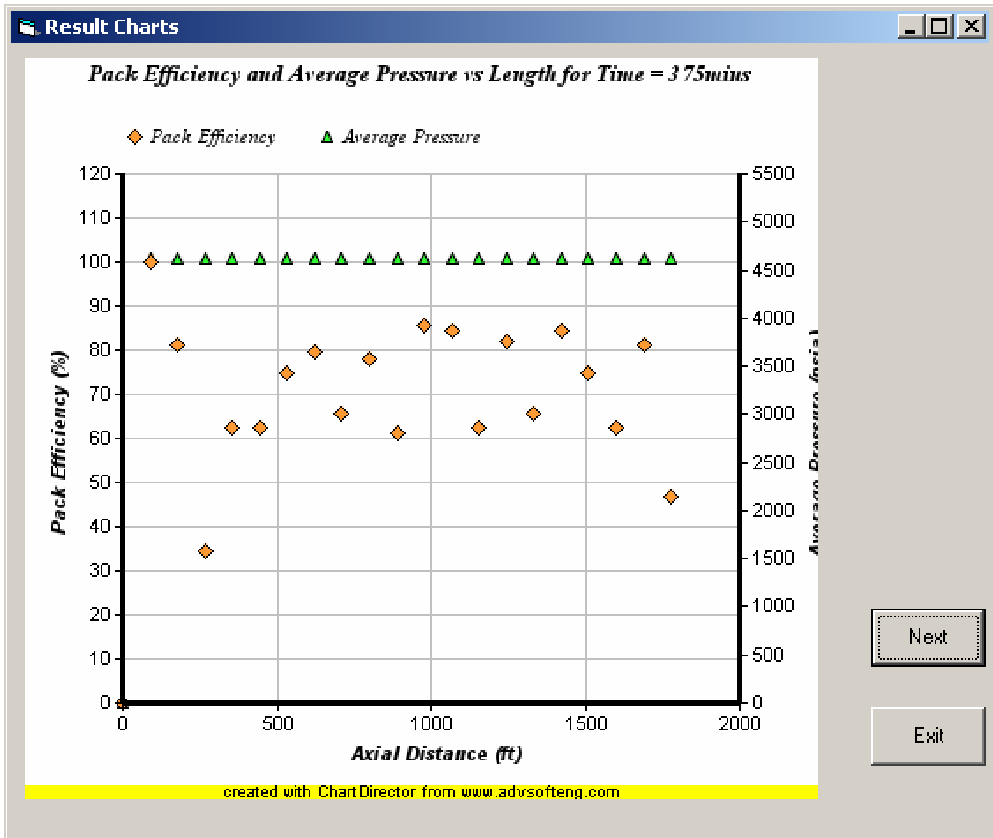


Figure A- 5.9: Example 1 Result Charts Screen at Time = 375min

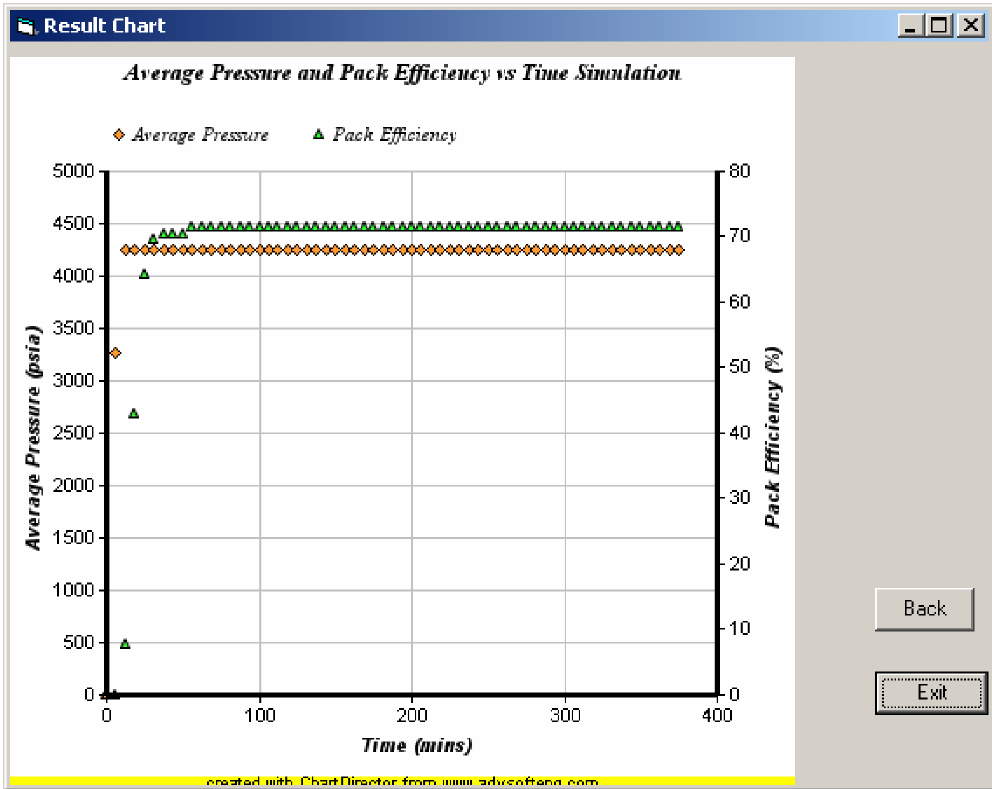


Figure A- 5.10: Example 1 Final Result Charts Screen

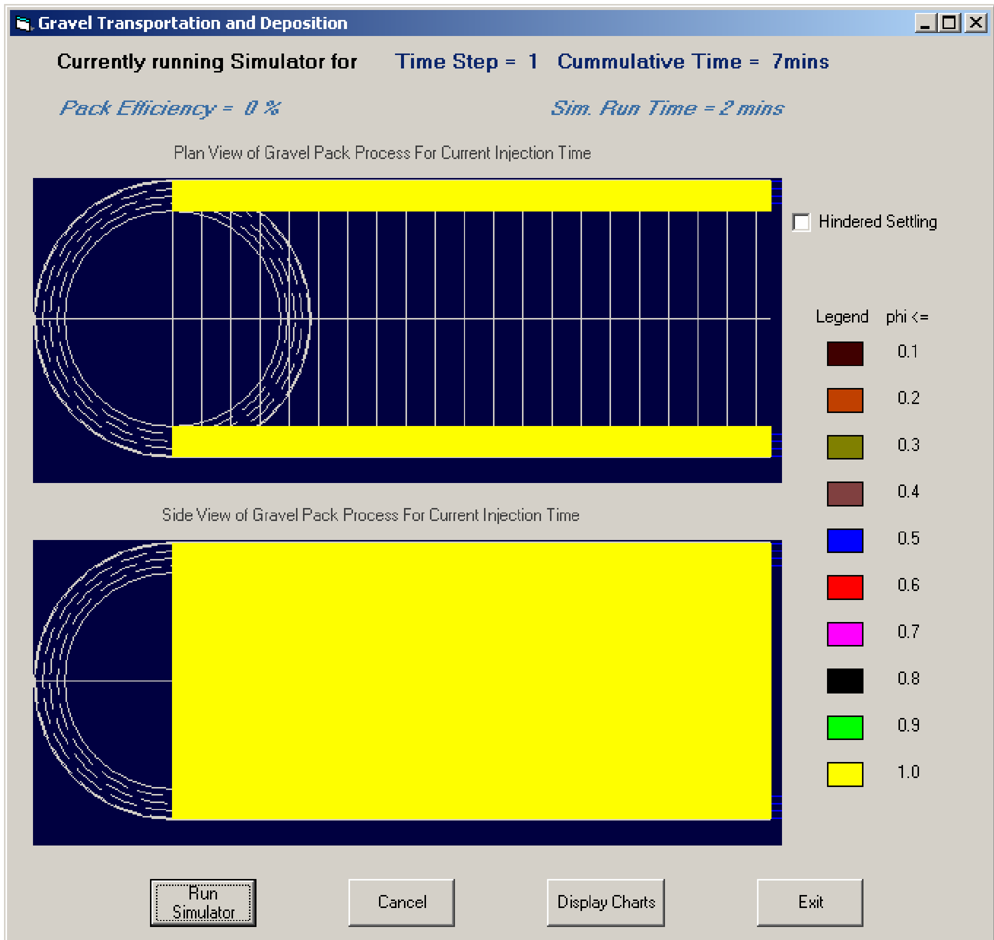


Figure A- 5.11: Example 1 Case 2 Simulation for time step = 1

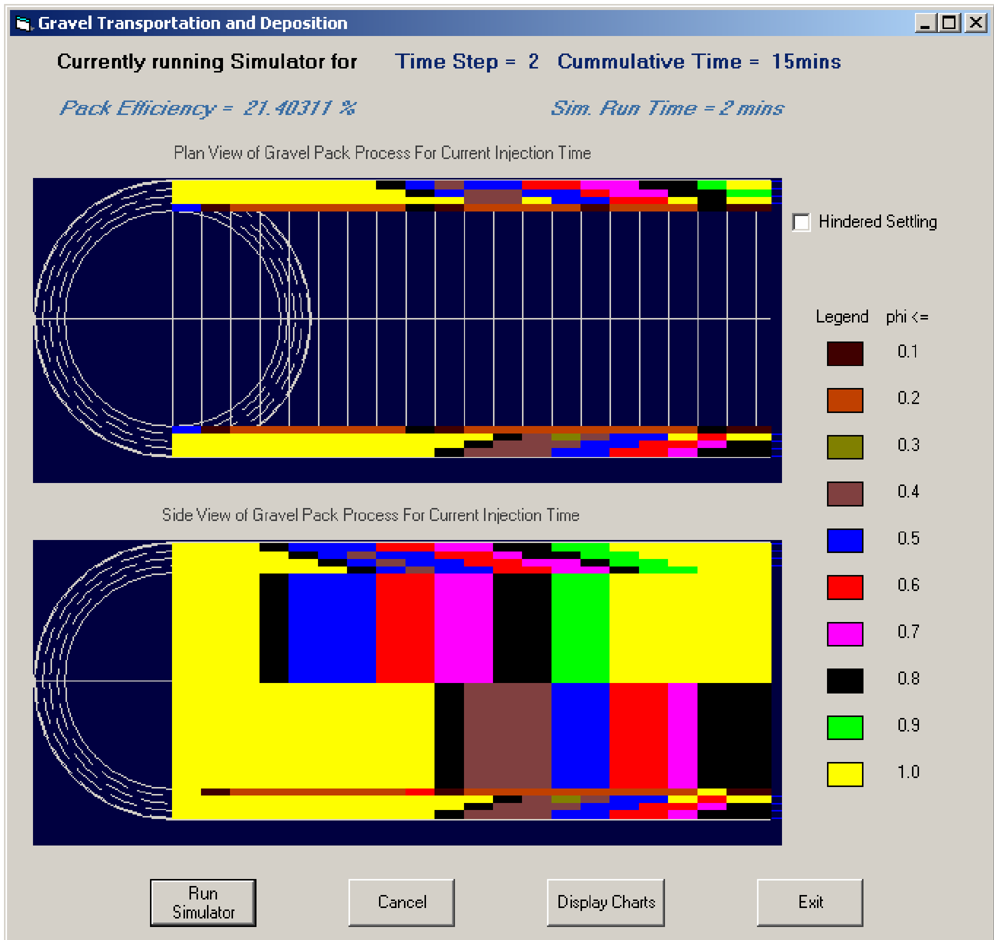


Figure A- 5.12: Example 1 Case 2 Simulation for time step = 2

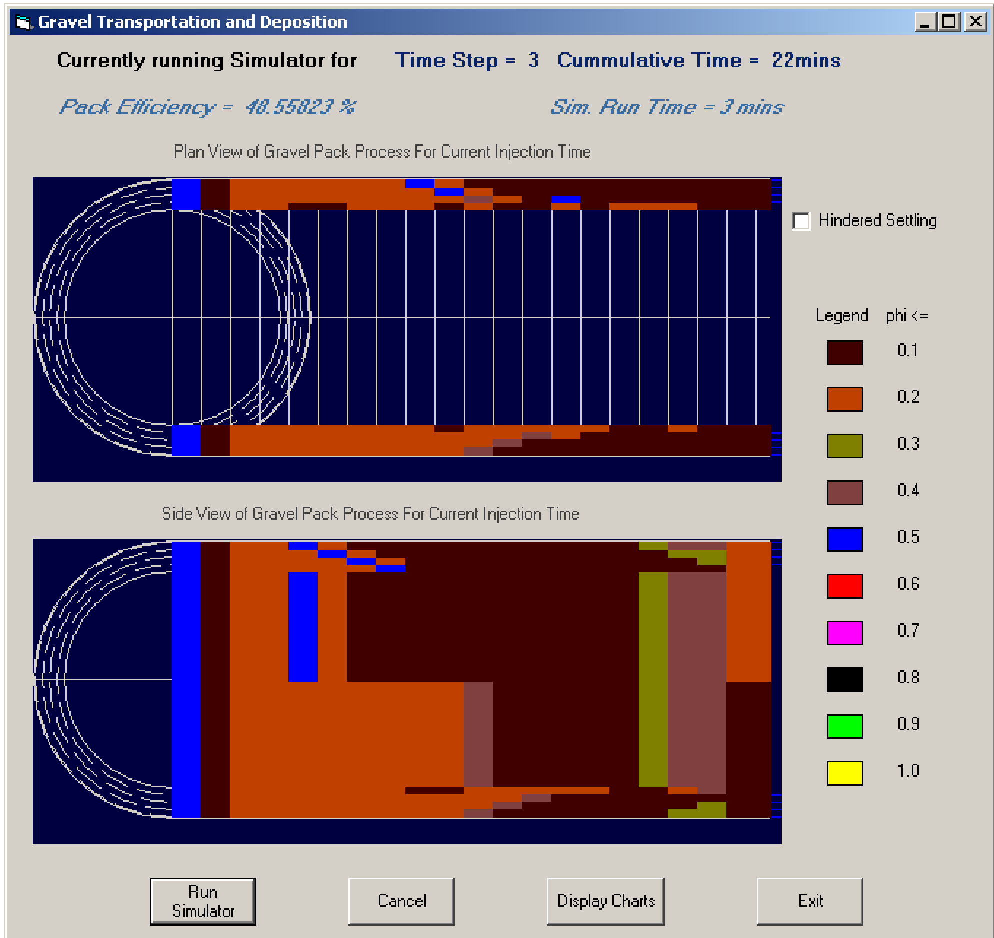


Figure A- 5.13: Example 1 Case 2 Simulation for time step = 3

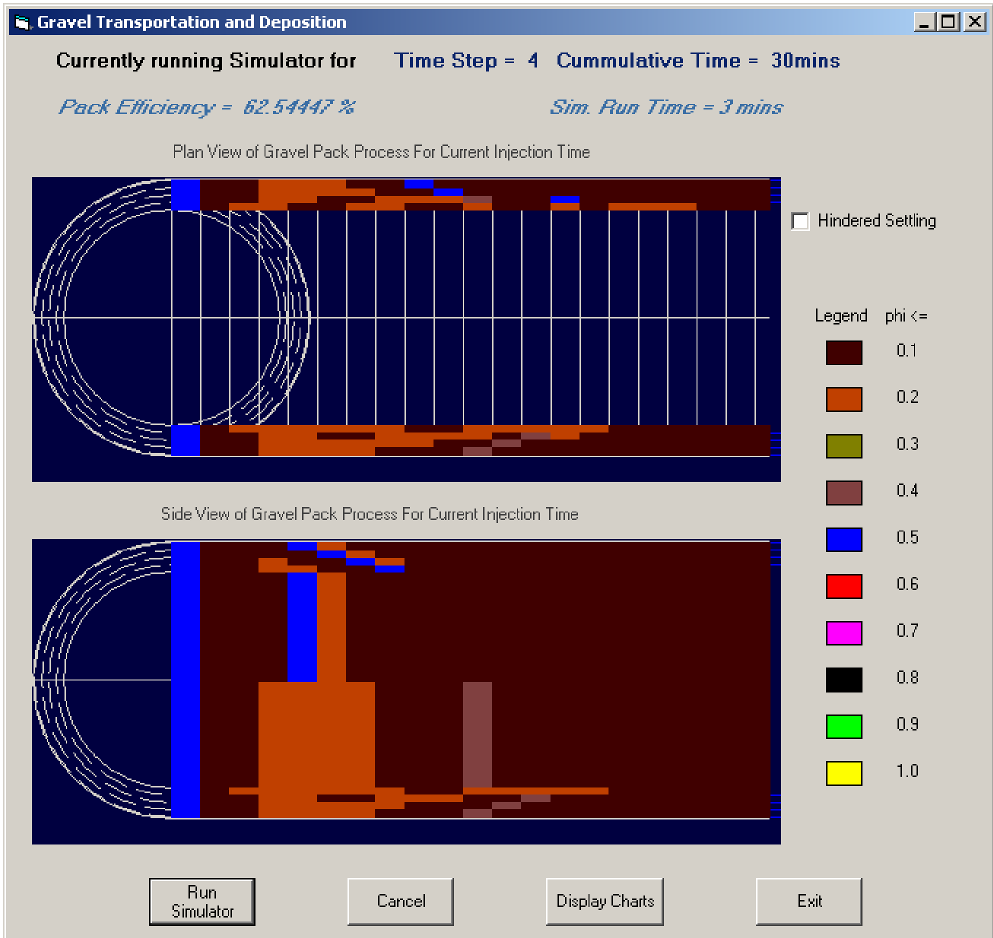


Figure A- 5.14: Example 1 Simulation in Progress for time step = 4

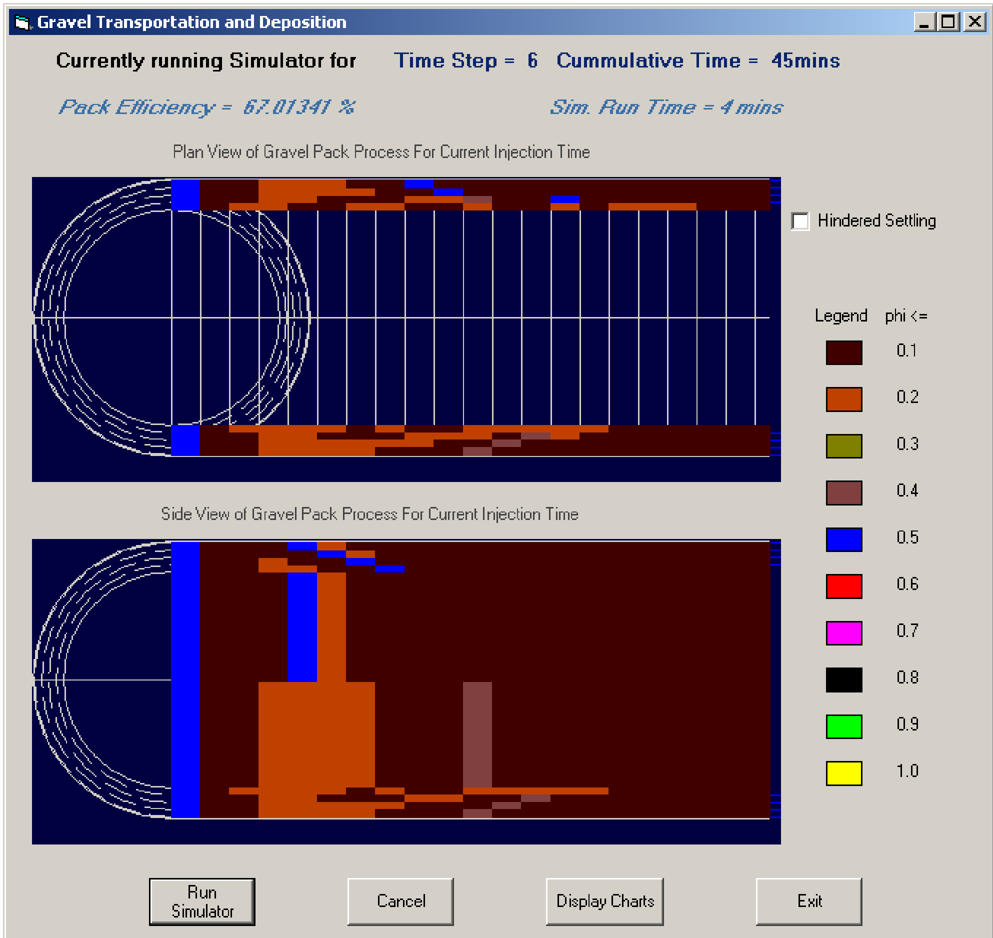


Figure A- 5.15: Example 1 Case 2 Simulation for time step = 6

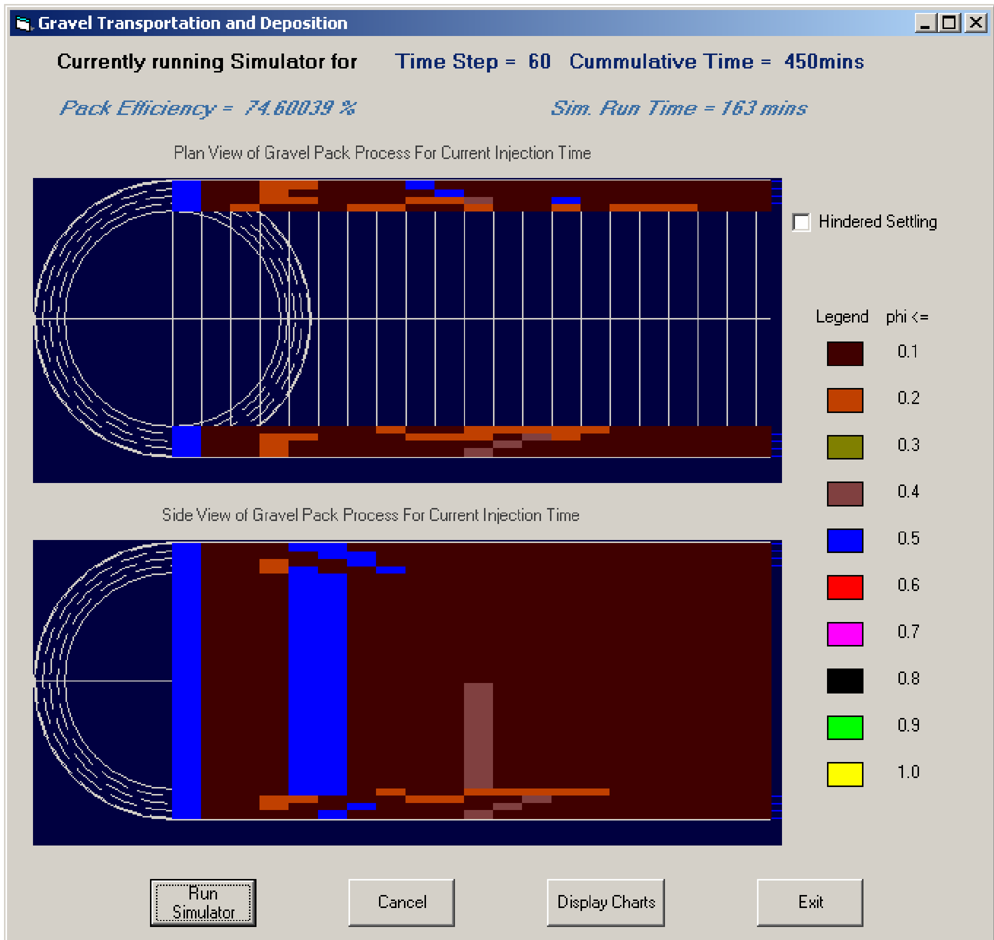


Figure A- 5.16: Example 1 Case 2 Simulation for time step = 60

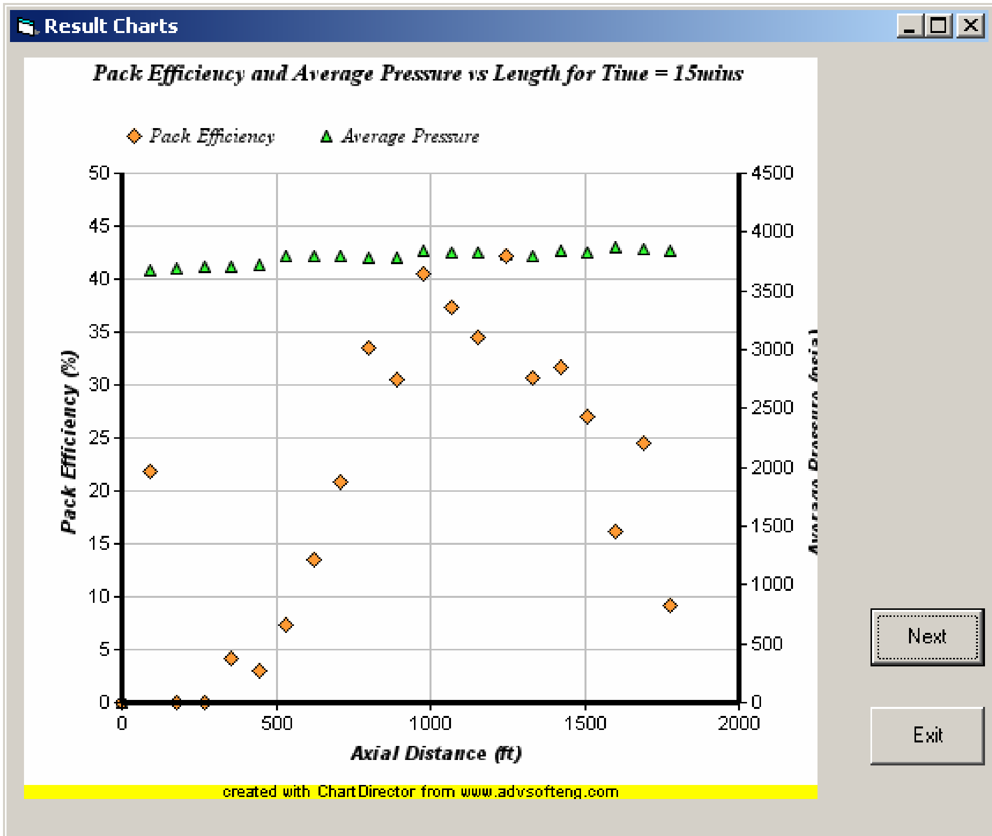


Figure A- 5.17: Example 1 Case 2 Charts Screen at Time = 15min

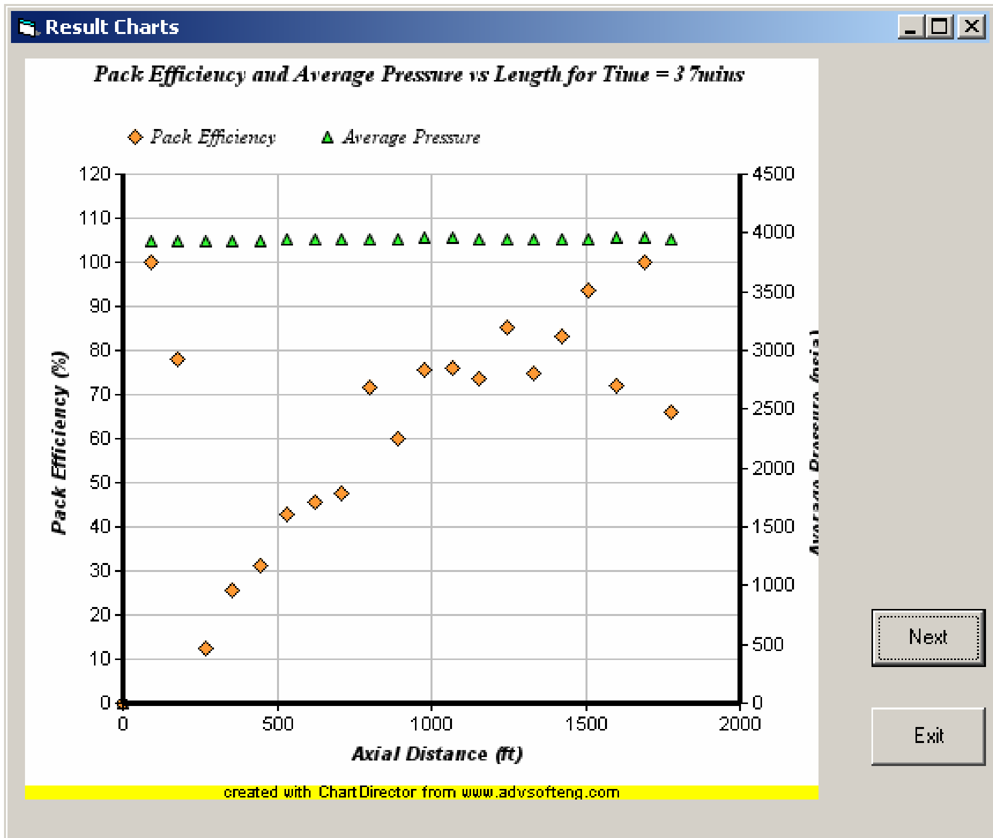


Figure A- 5.18: Example 1 Case 2 Charts Screen at Time = 37min

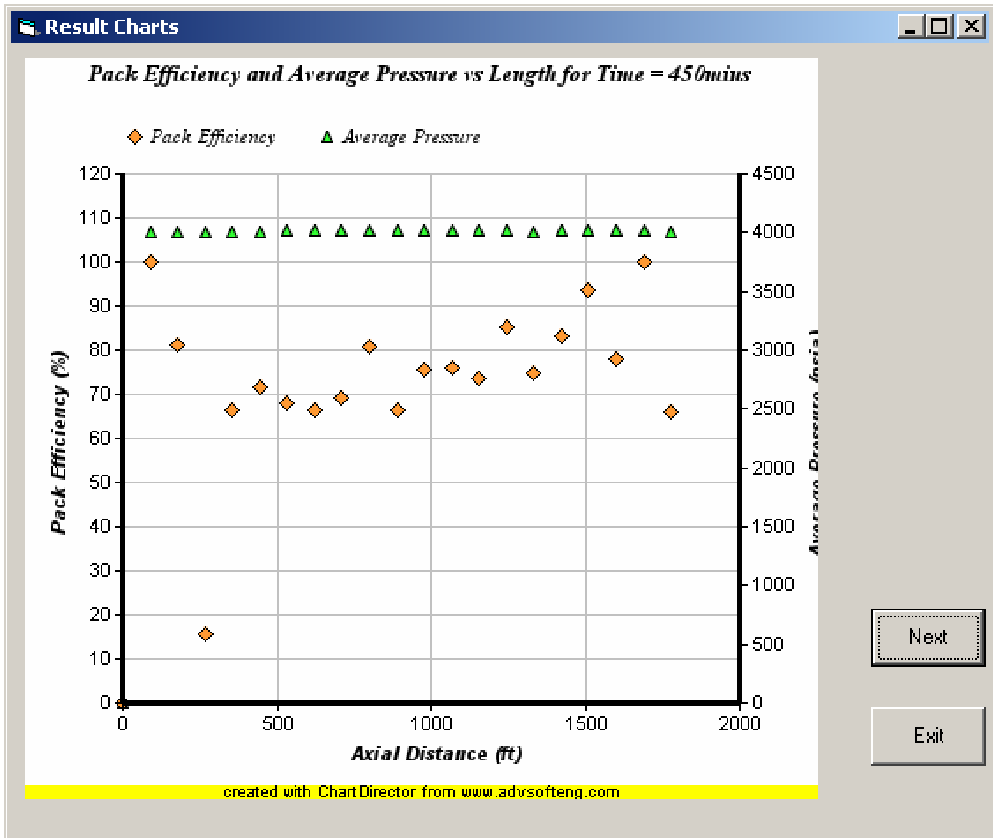


Figure A- 5.19: Example 1 Case 2 Charts Screen at Time = 450min

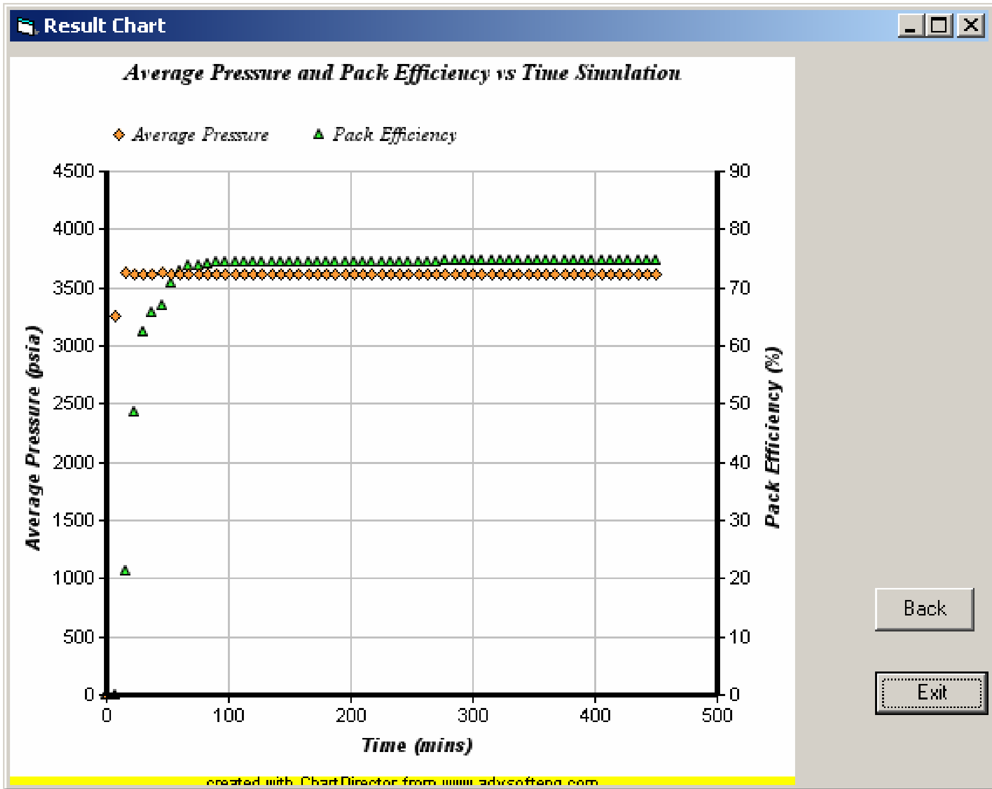


Figure A- 5.20: Example 1 Case 2 Final Result Charts Screen

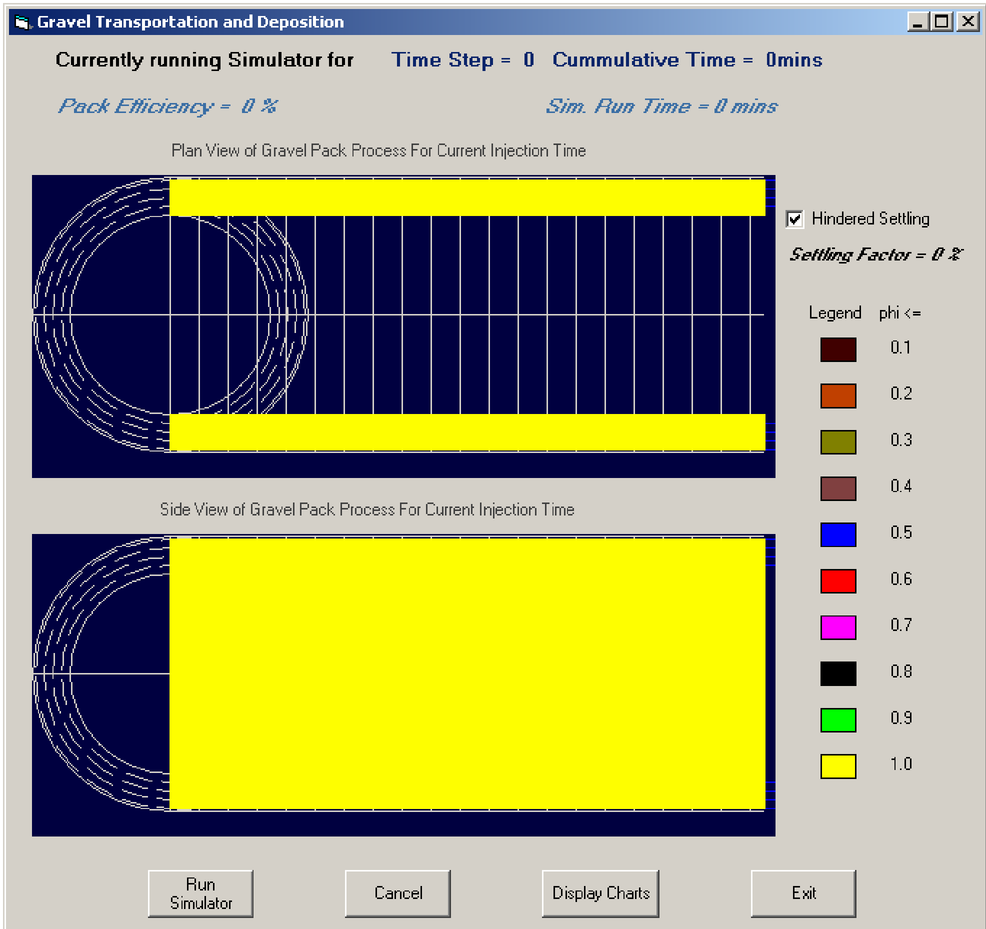


Figure A- 5.21: Example 2 Simulation in Progress for time step = 0

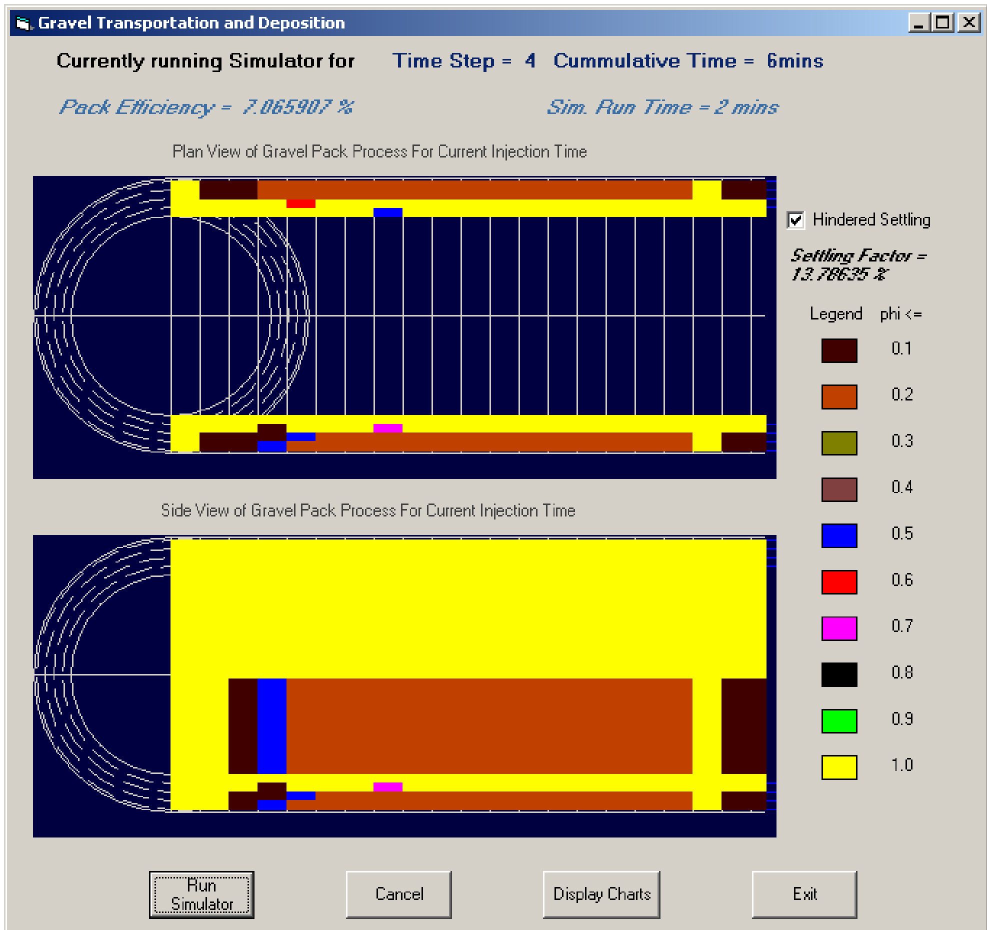


Figure A- 5.22: Example 2 Simulation in Progress for time step = 4

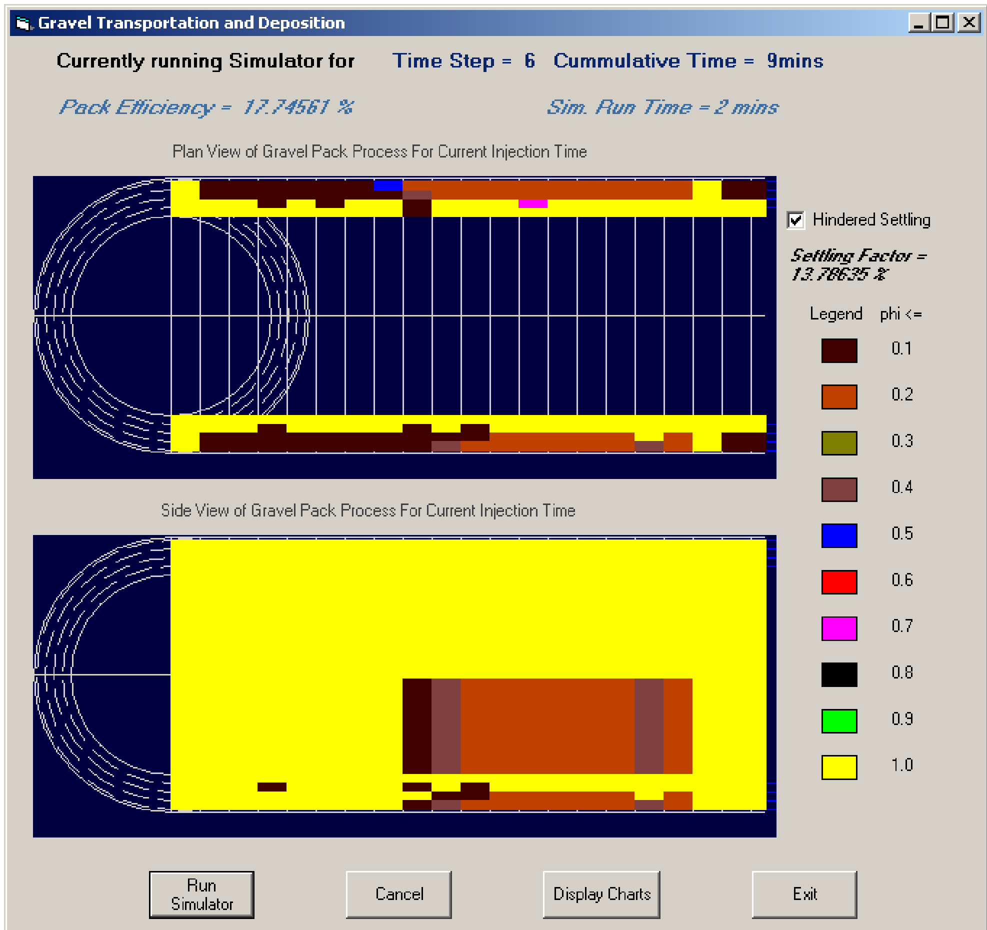


Figure A- 5.23: Example 2 Simulation in Progress for time step = 6

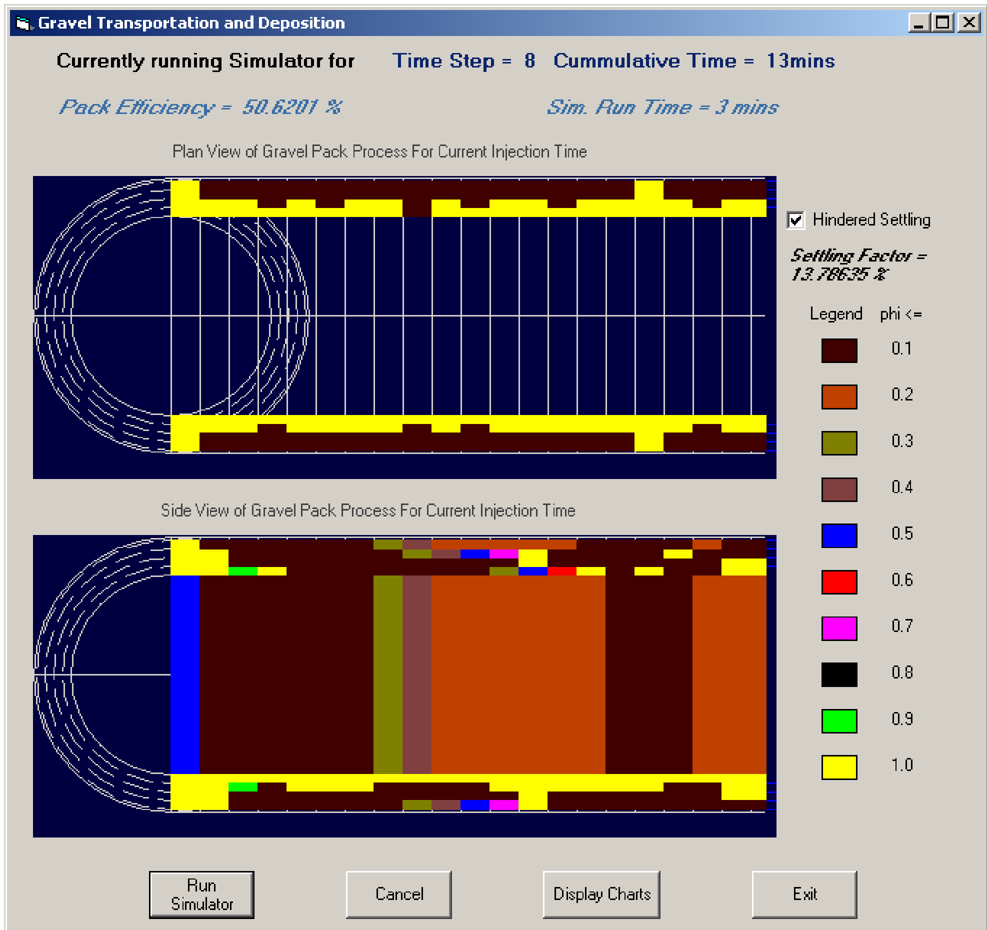


Figure A- 5.24: Example 2 Simulation in Progress for time step = 8

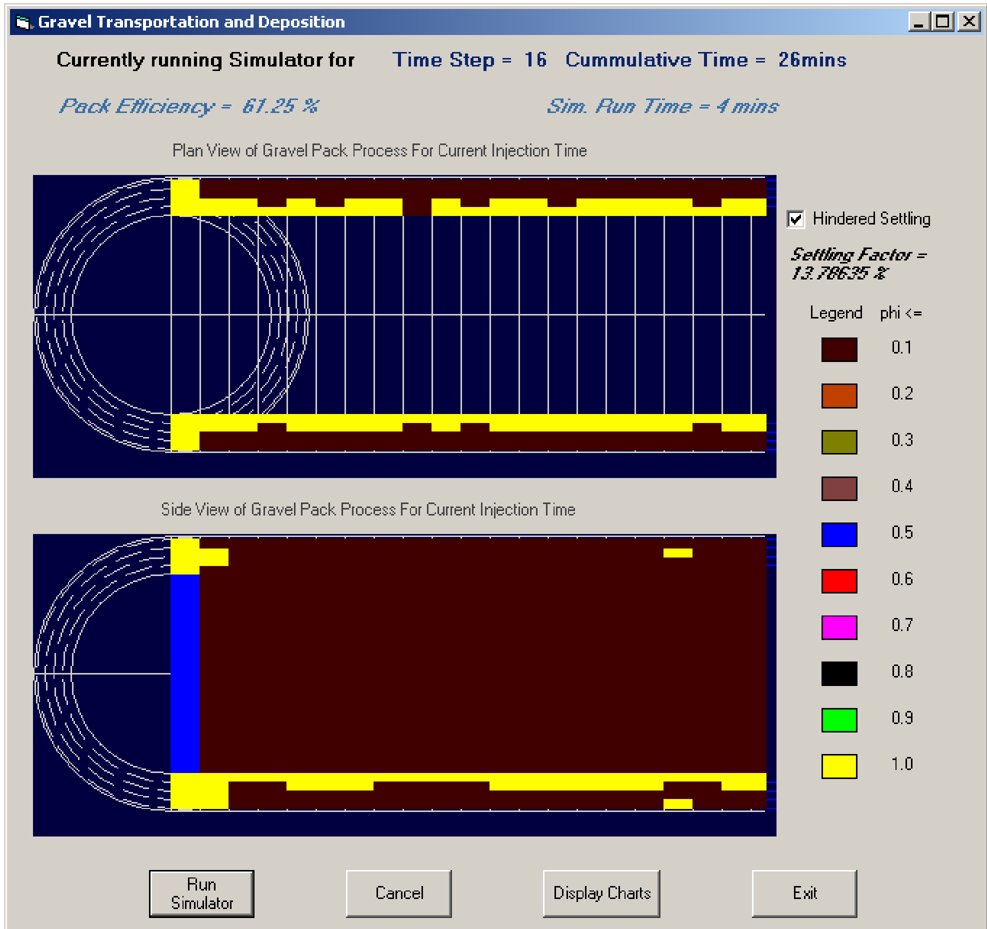


Figure A- 5.25: Example 2 Simulation in Progress for time step = 16

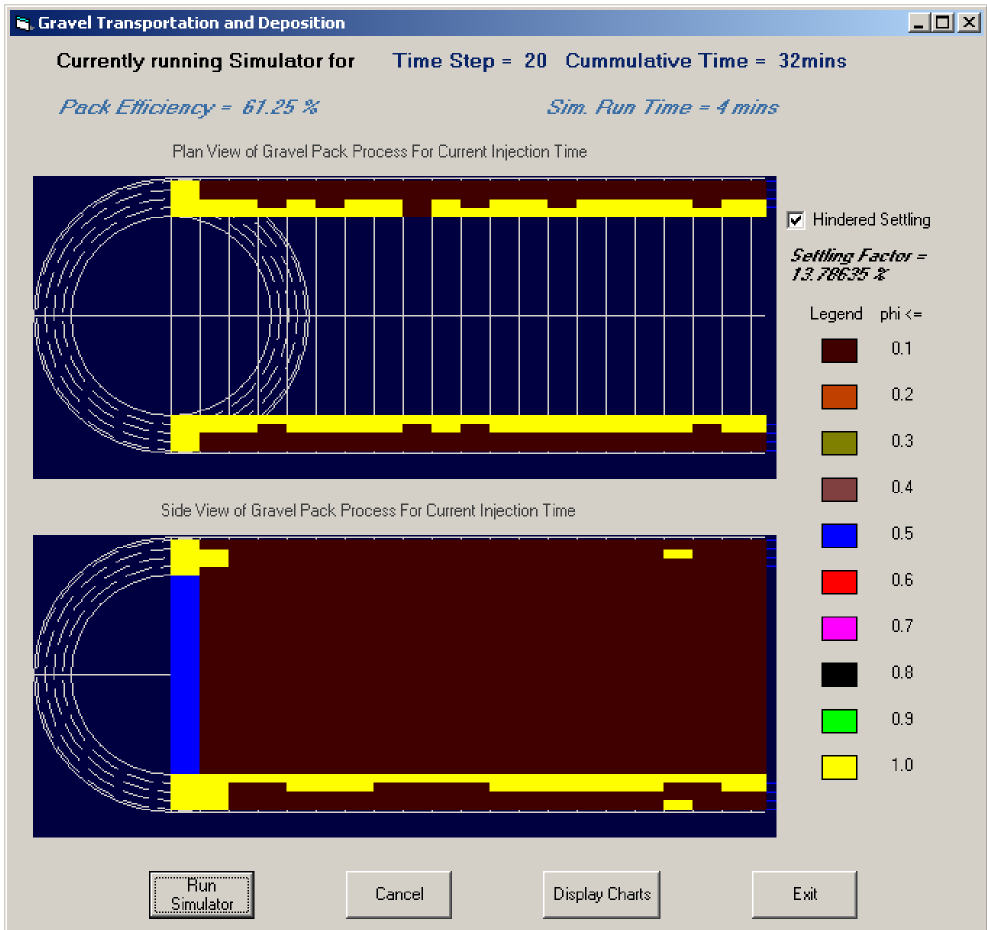


Figure A- 5.26: Example 2 Simulation in Progress for time step = 20

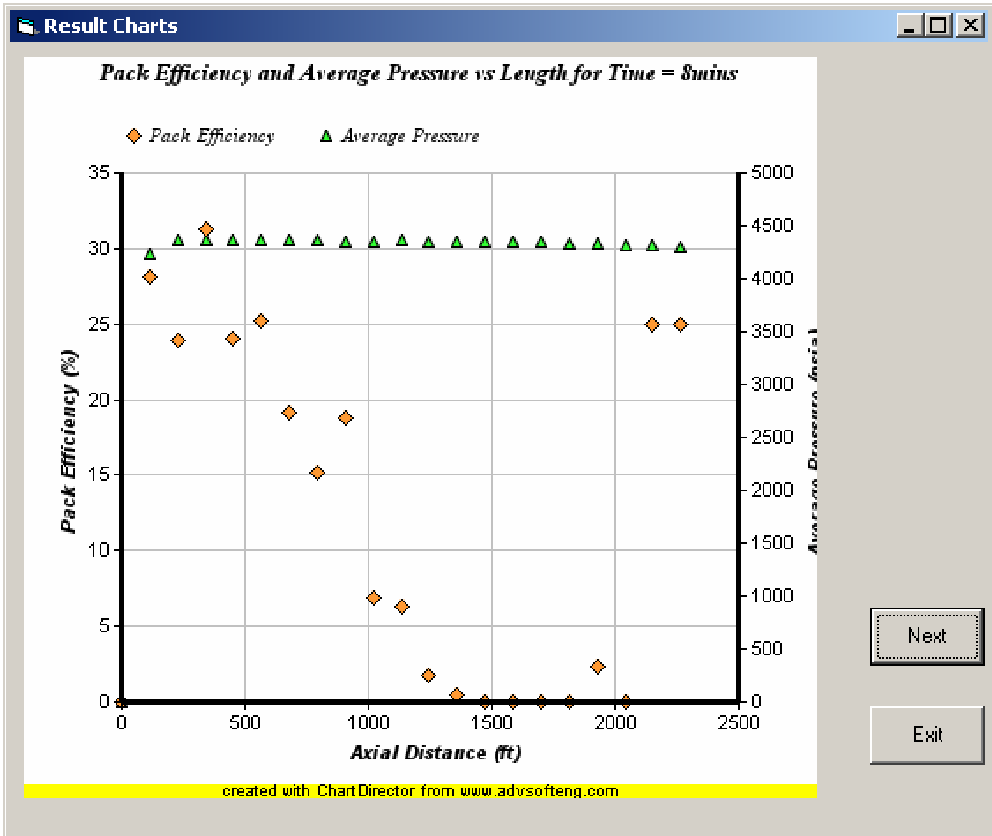


Figure A- 5.27: Example 2 Result Charts Screen at Time = 8min

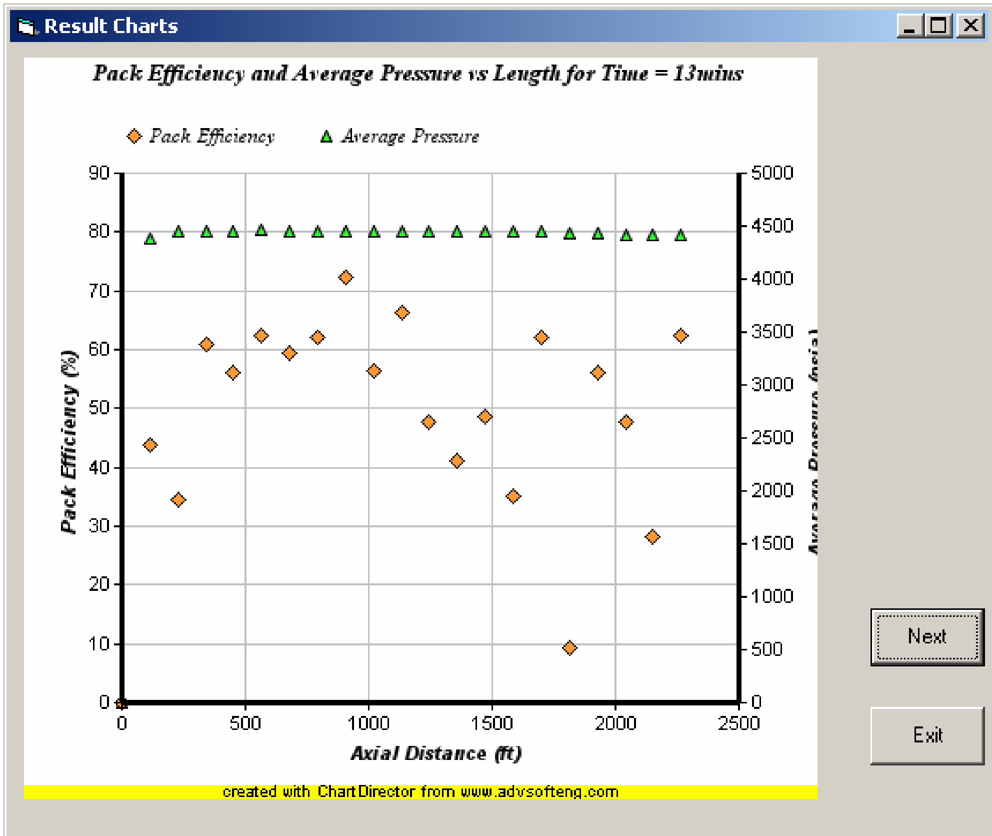


Figure A- 5.28: Example 2 Result Charts Screen at Time = 13min

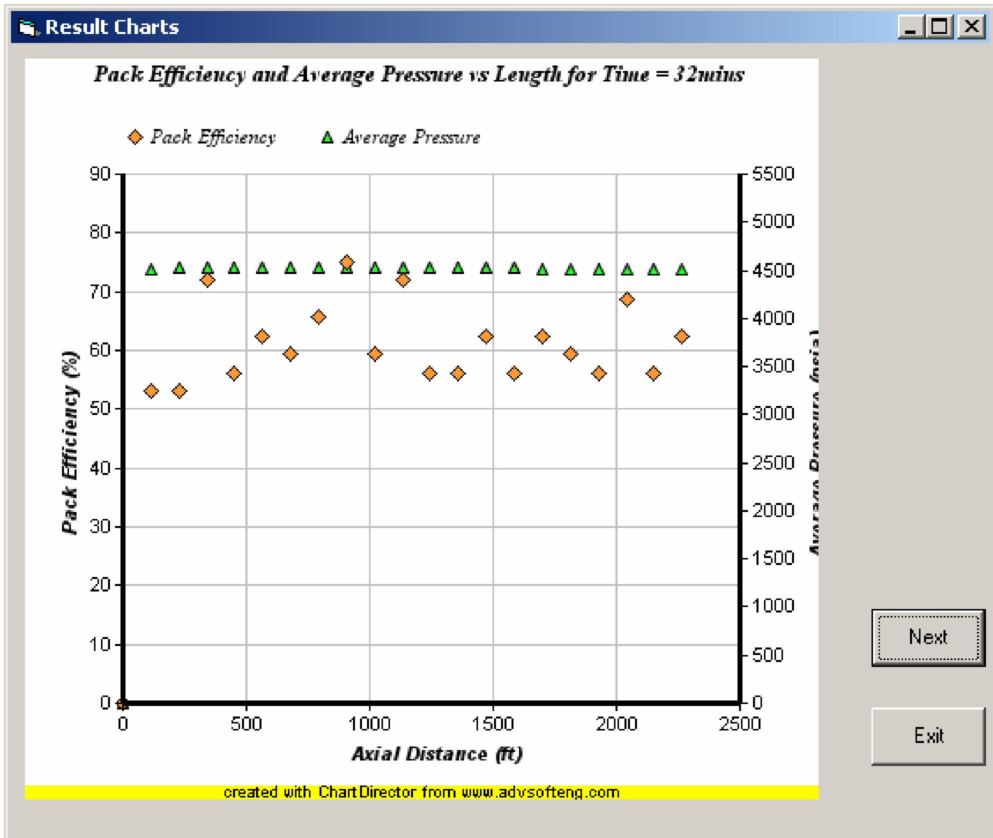


Figure A- 5.29: Example 2 Result Charts Screen at Time = 32min

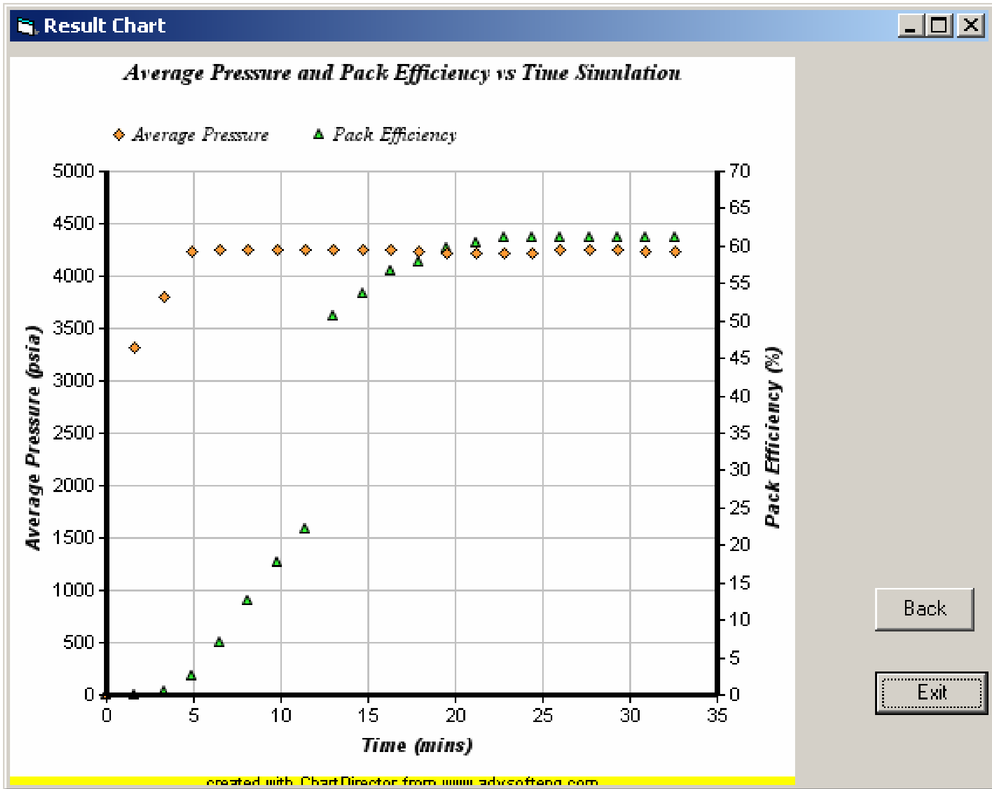


Figure A- 5.30: Example 2 Final Result Charts Screen

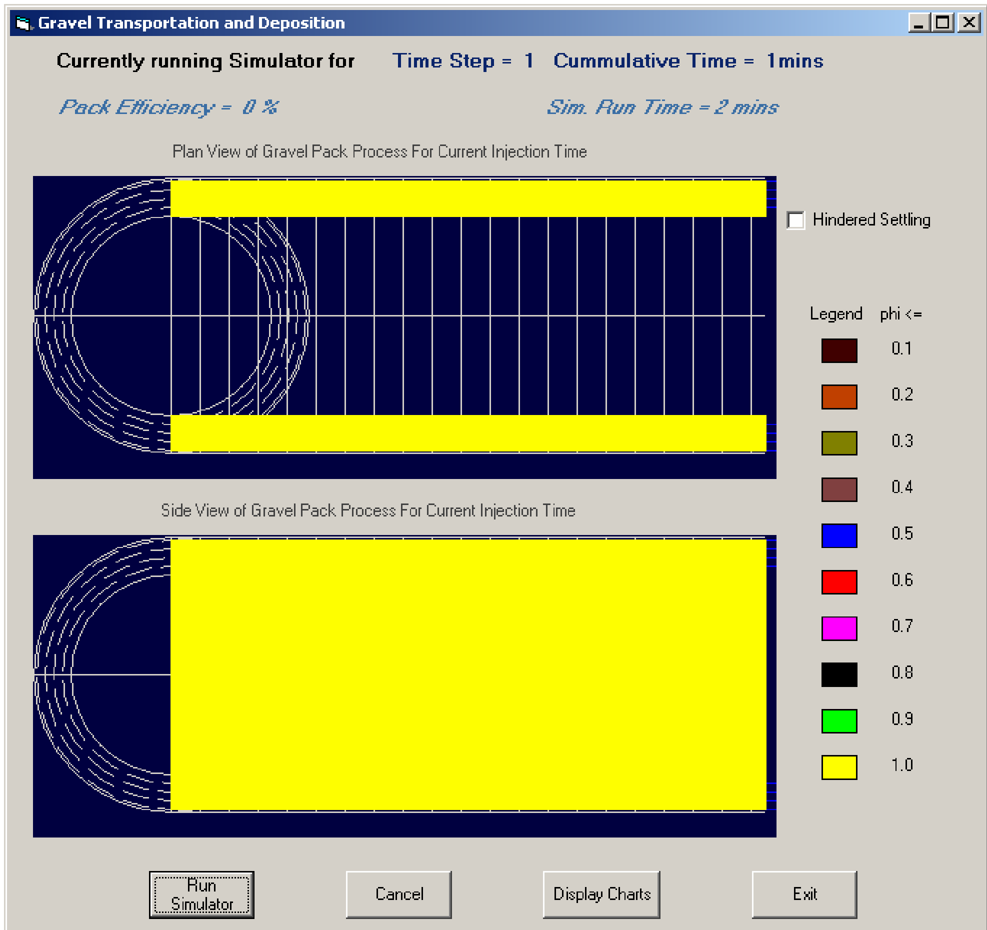


Figure A- 5.31: Example 2 Case 2 Simulation for time step = 1

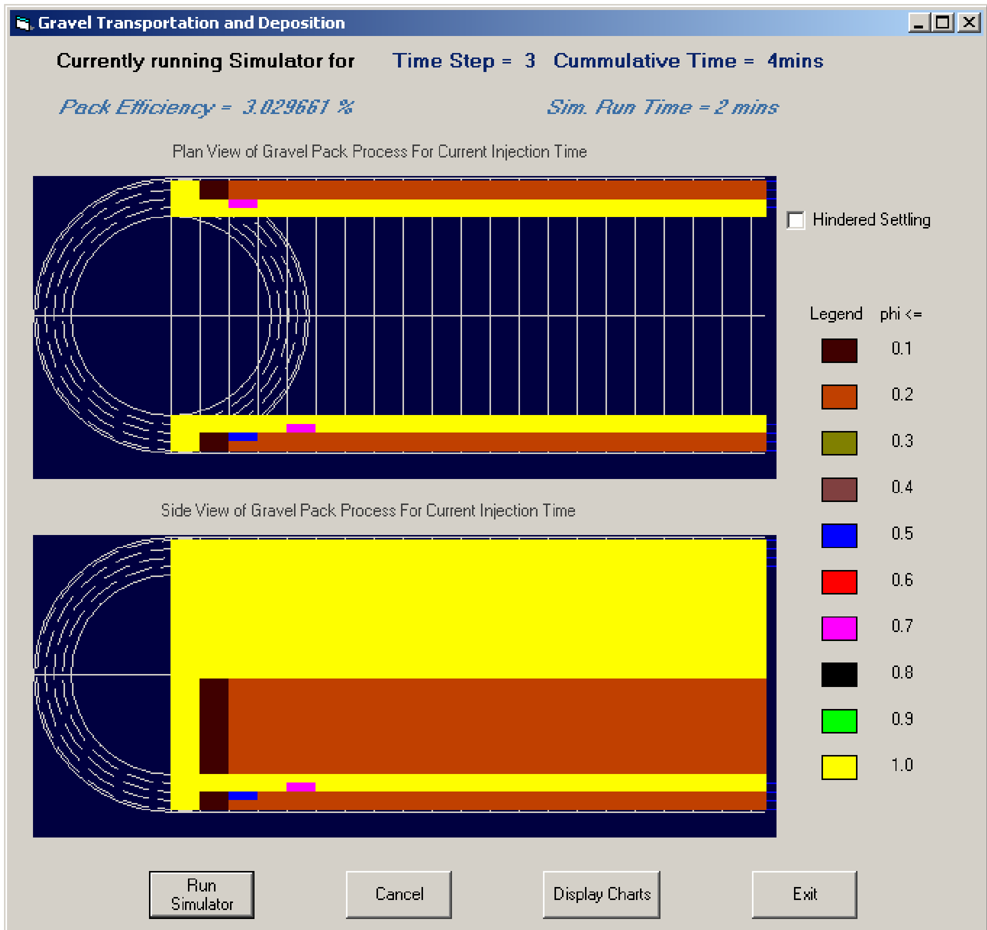


Figure A- 5.32: Example 2 Case 2 Simulation for time step = 3

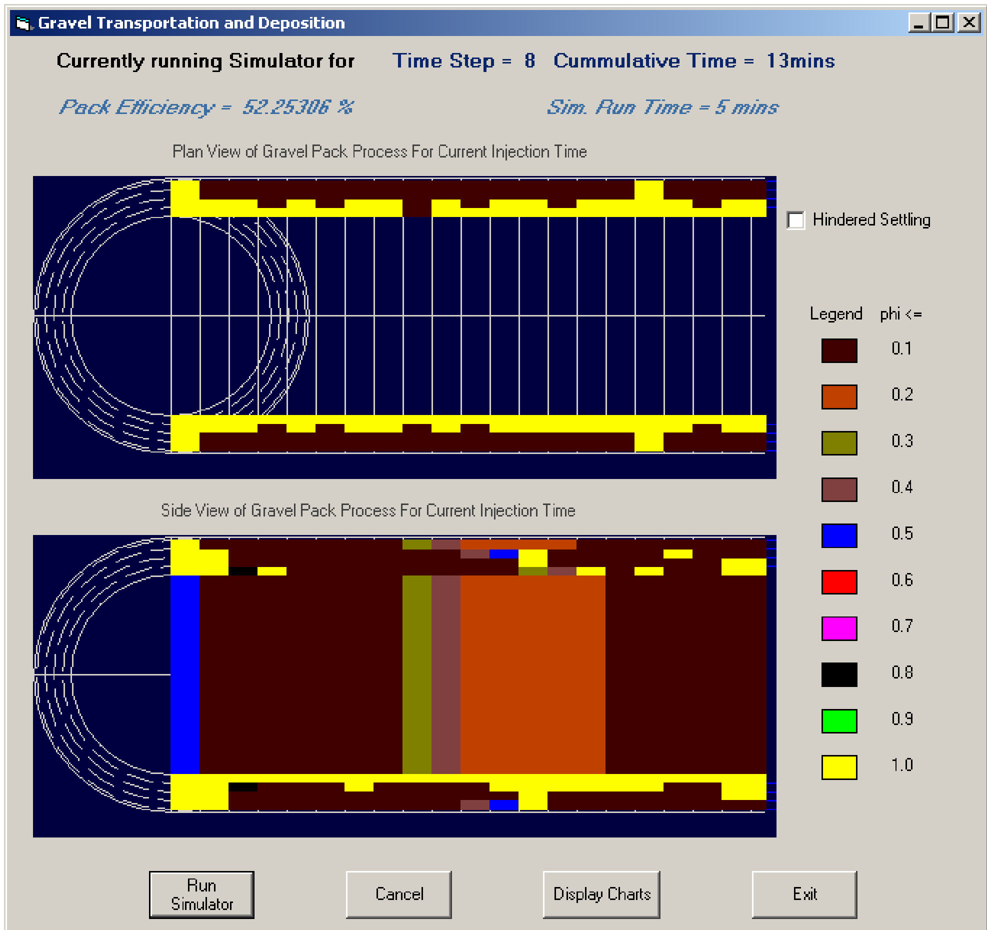


Figure A- 5.33: Example 2 Case 2 Simulation for time step = 8

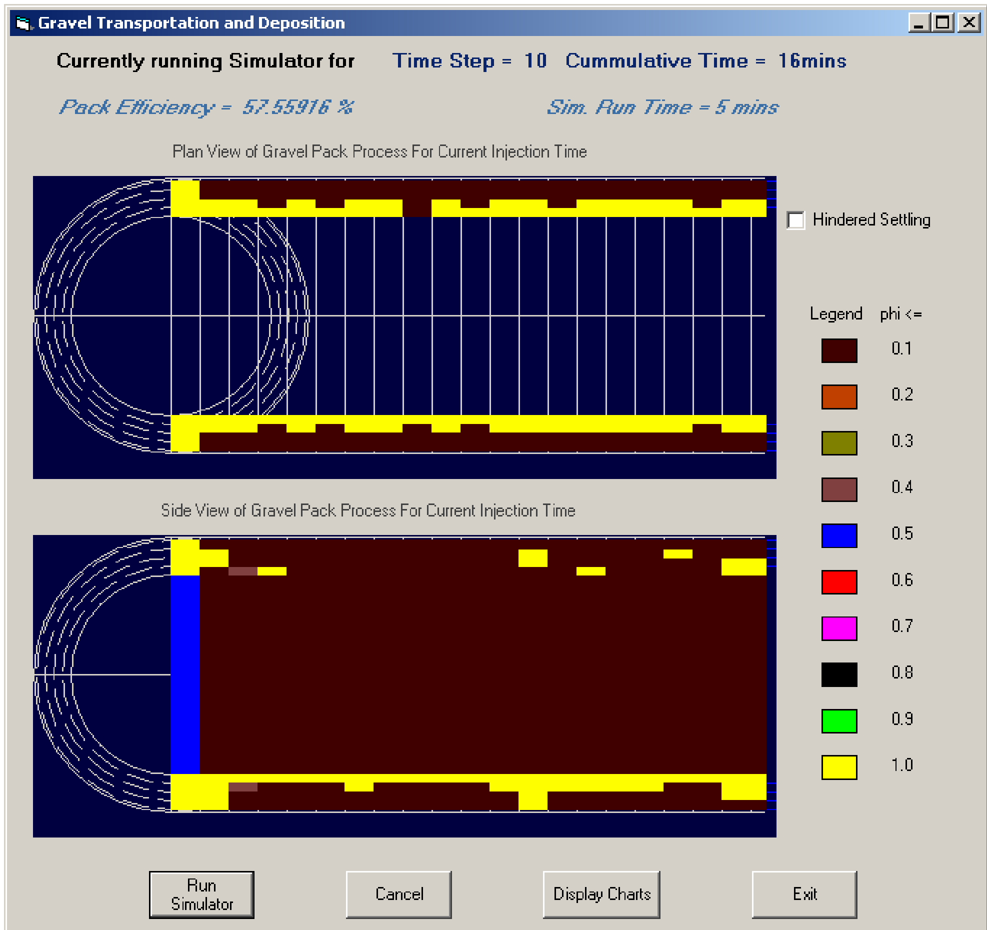


Figure A- 5.34: Example 2 Case 2 Simulation for time step = 10

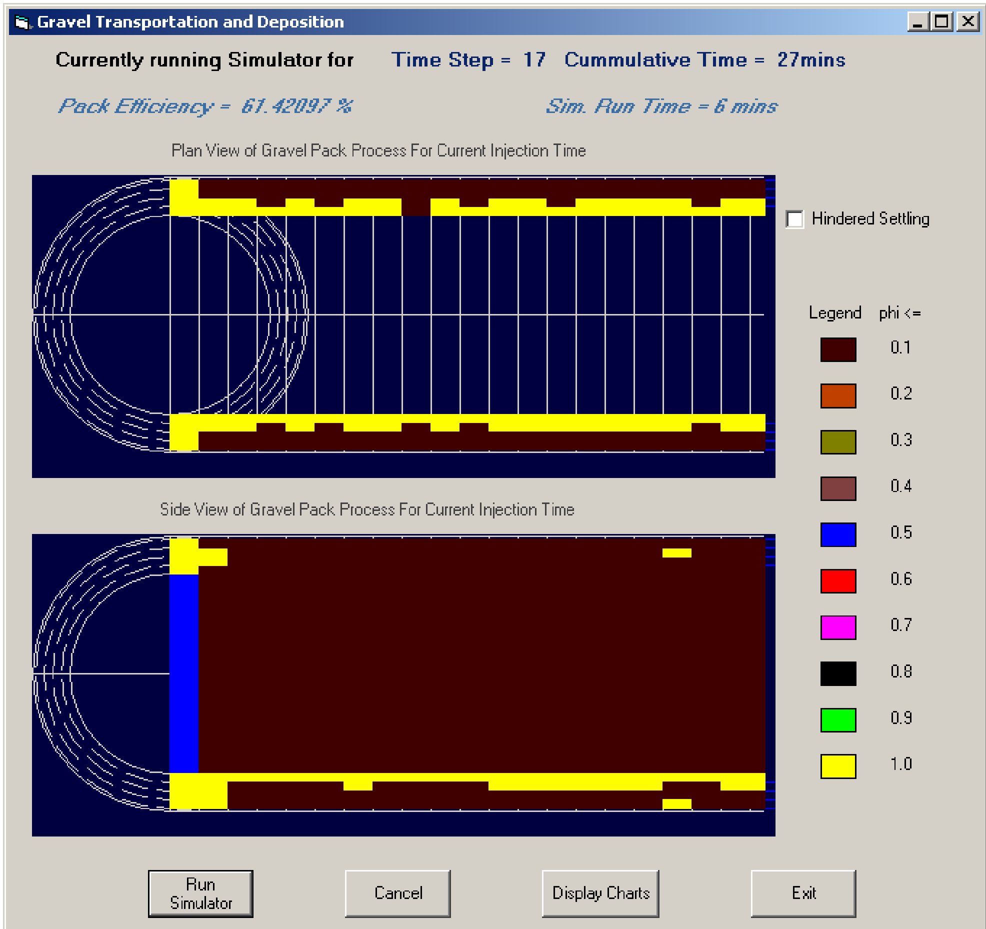


Figure A- 5.35: Example 2 Case 2 Simulation for time step = 17

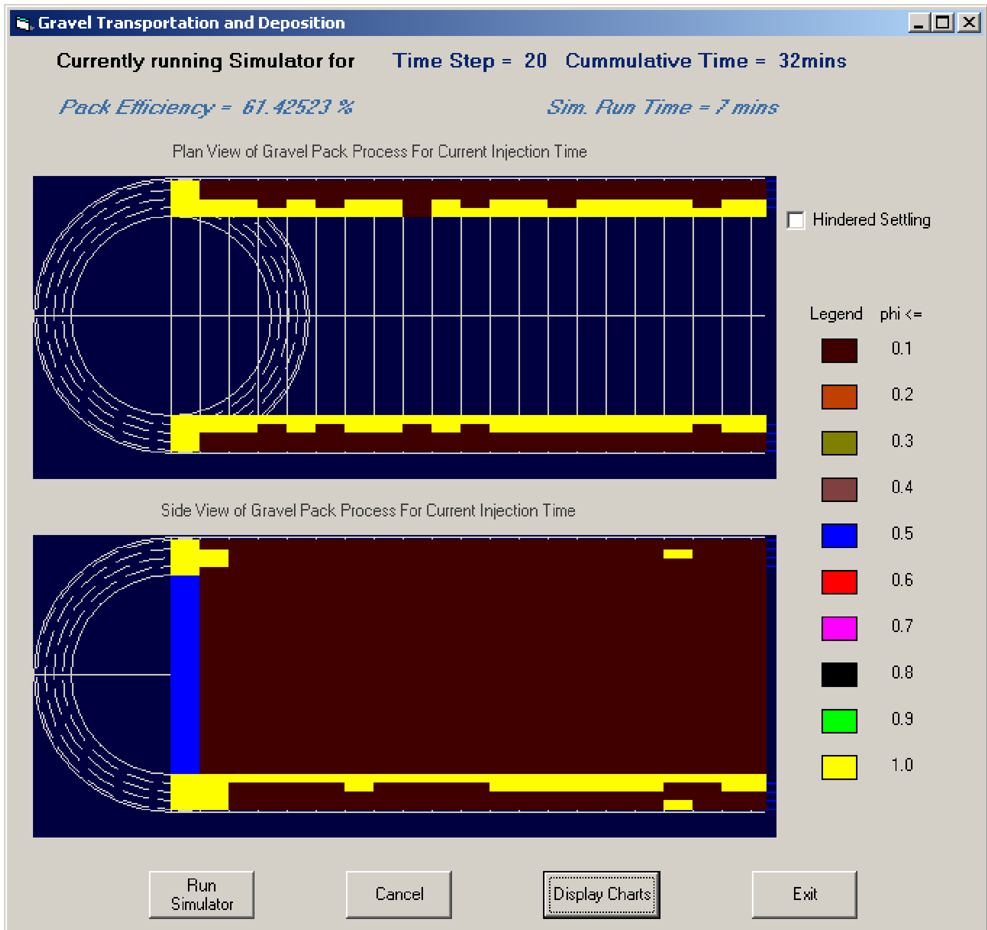


Figure A- 5.36: Example 2 Case 2 Simulation for time step = 20

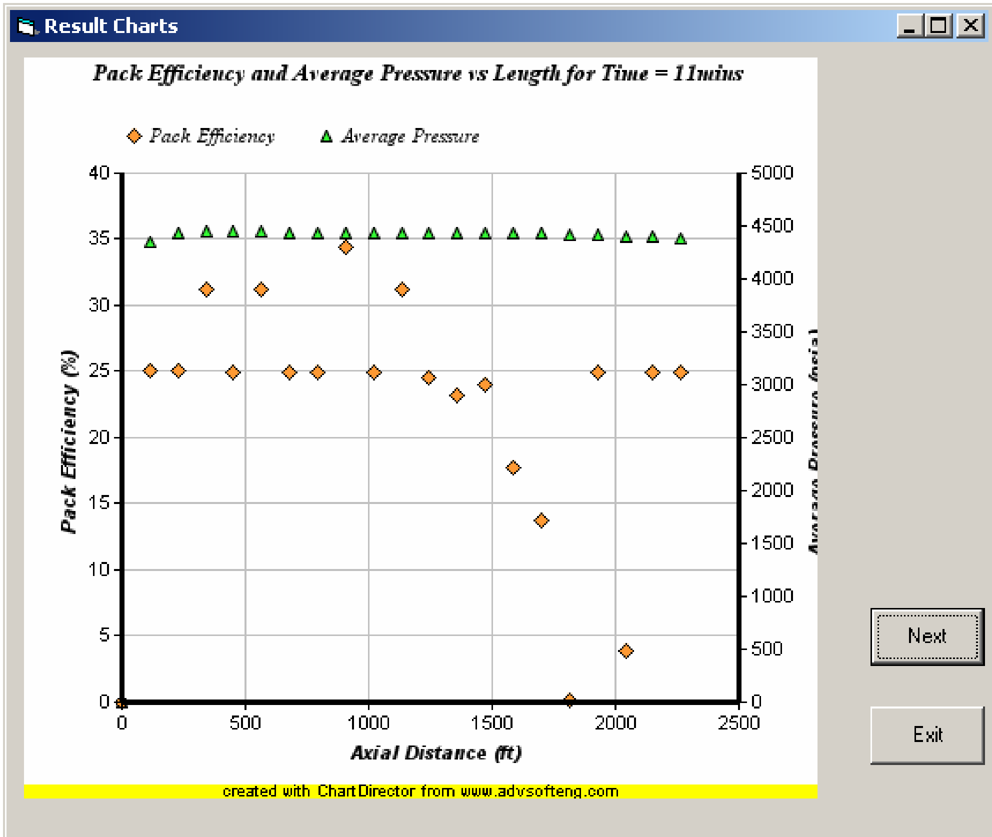


Figure A- 5.37: Example 2 Case 2 Charts Screen at Time = 11min

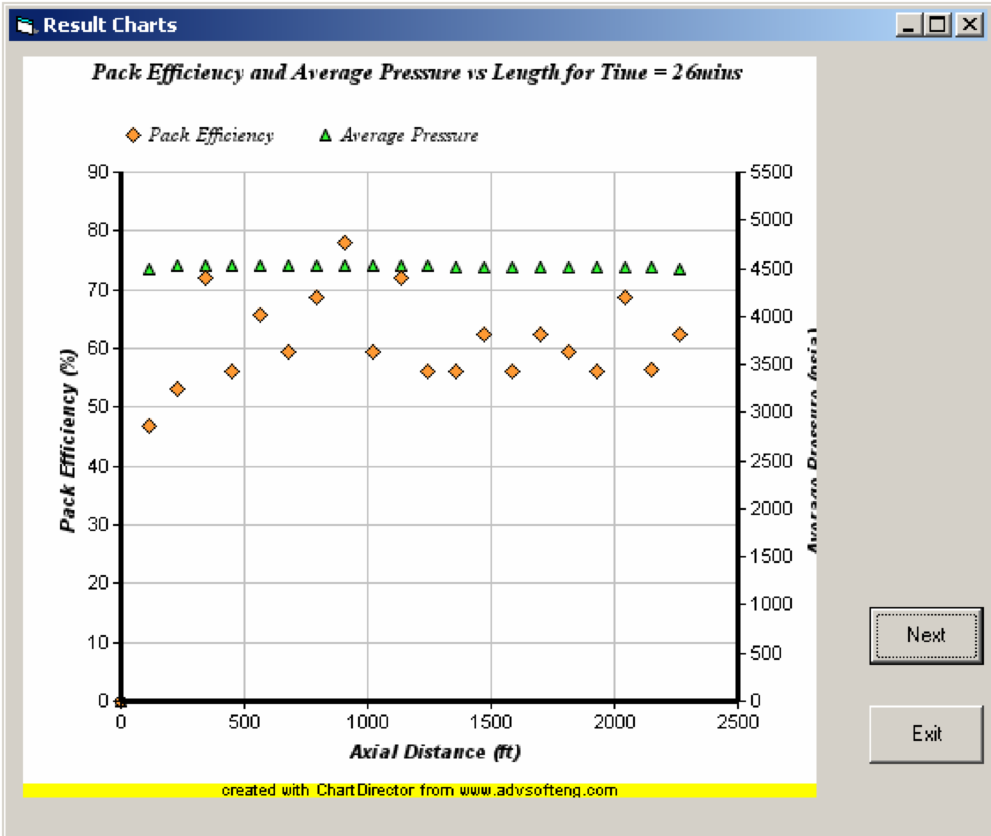


Figure A- 5.38: Example 2 Case 2 Charts Screen at Time = 26min

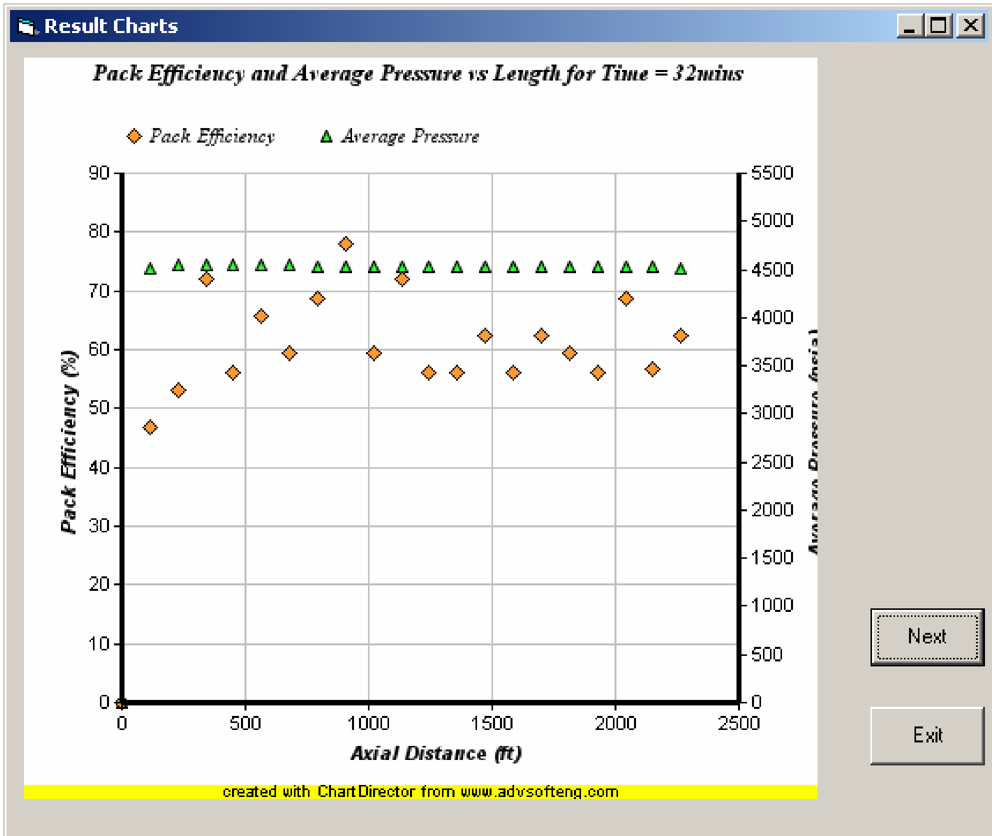


Figure A- 5.39: Example 2 Case 2 Charts Screen at Time = 32min

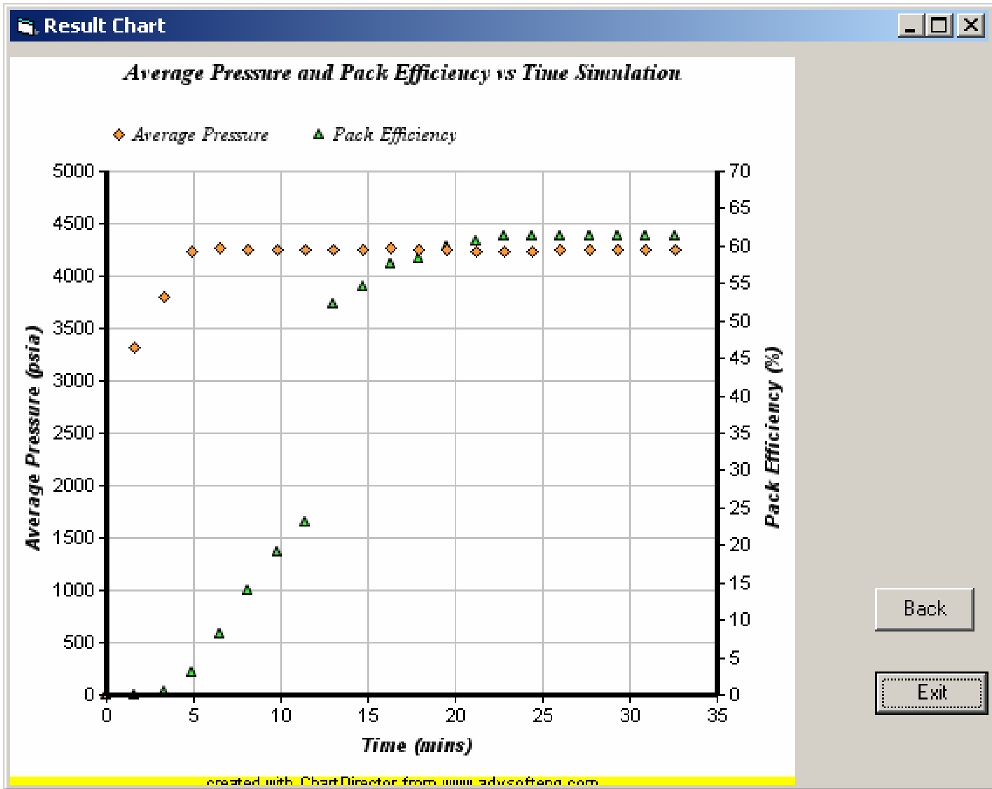


Figure A- 5.40: Example 2 Case 2 Final Result Charts Screen

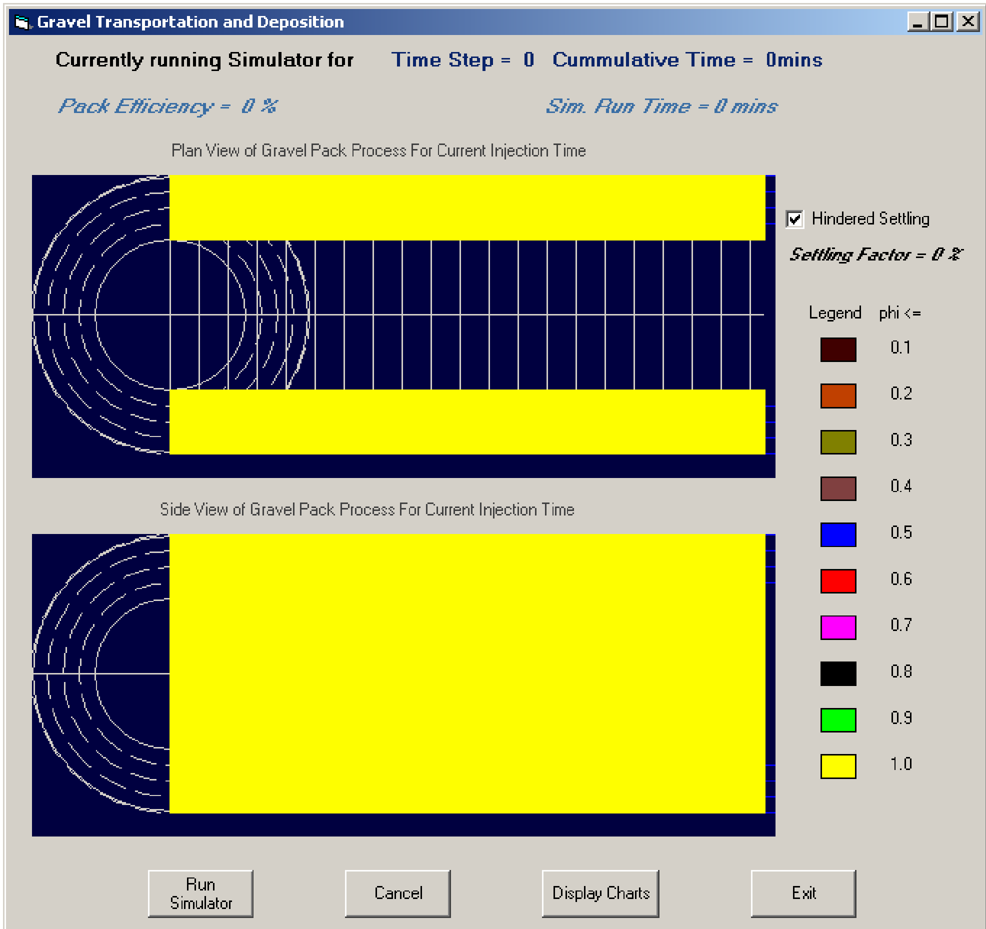


Figure A- 5.41: Example 3 Simulation in Progress for time step = 0

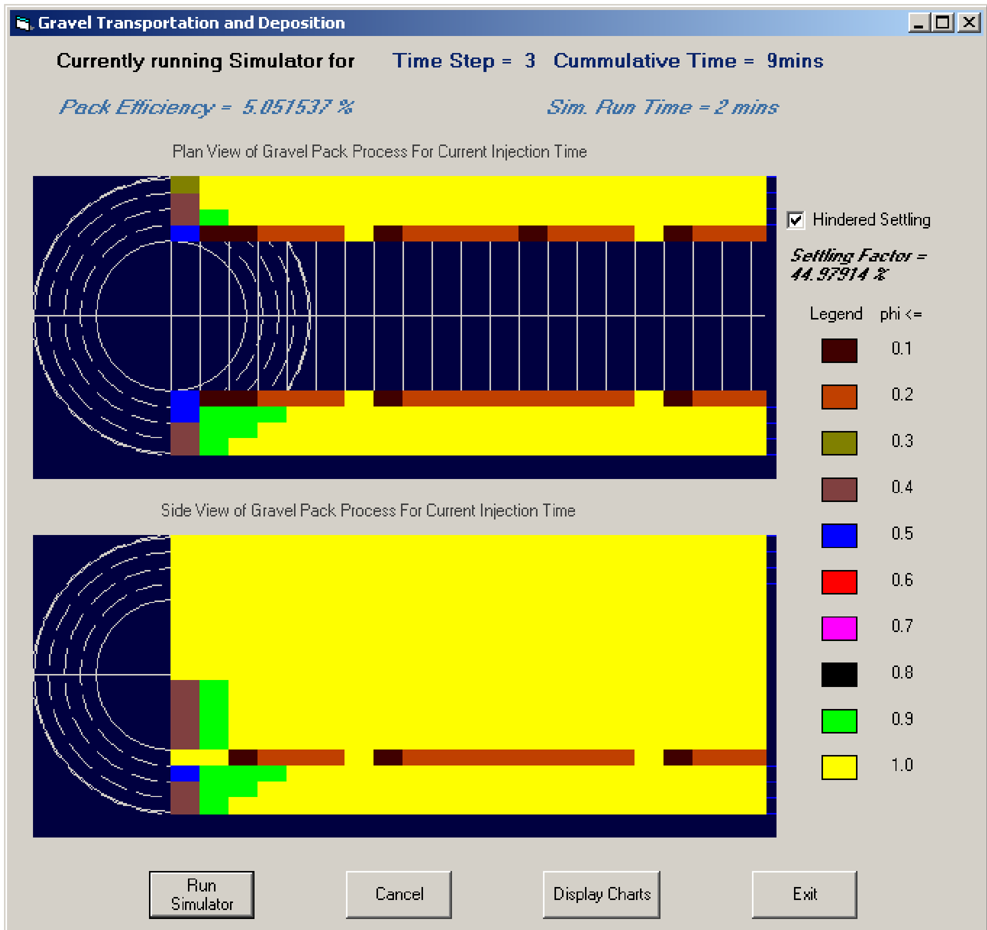


Figure A- 5.42: Example 3 Simulation in Progress for time step = 3

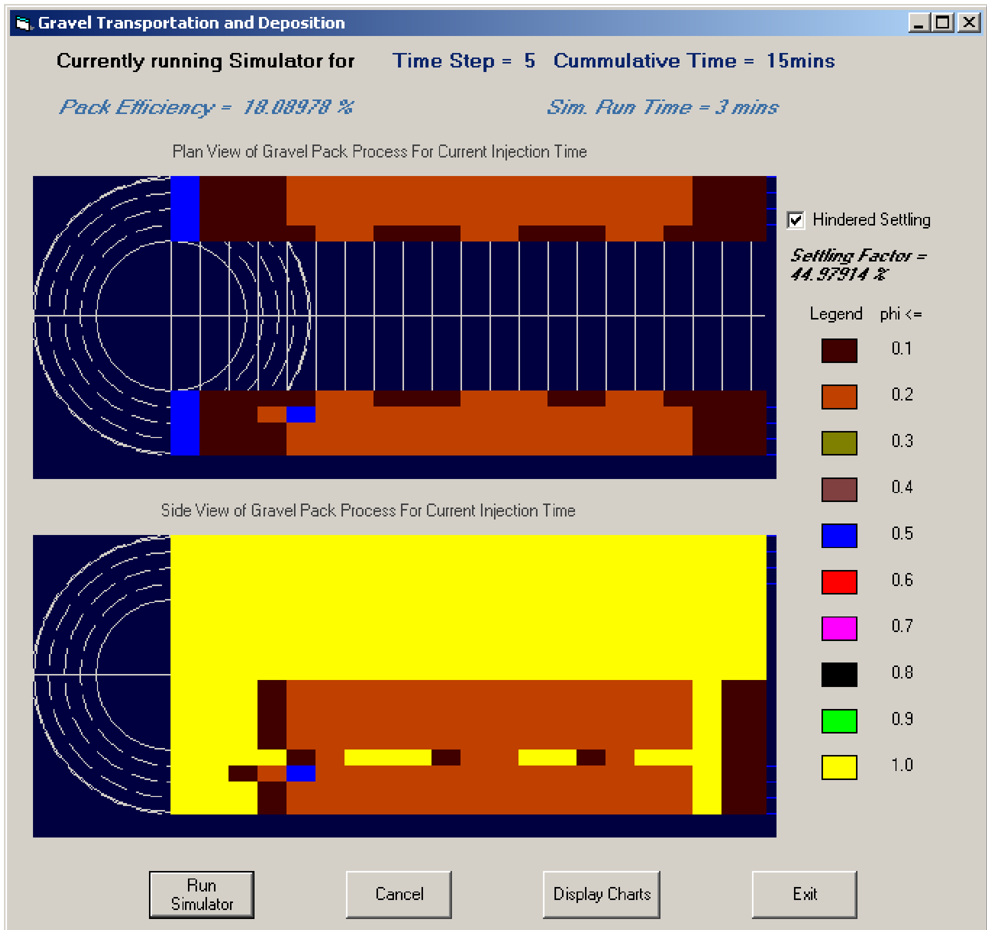


Figure A- 5.43: Example 3 Simulation in Progress for time step = 5

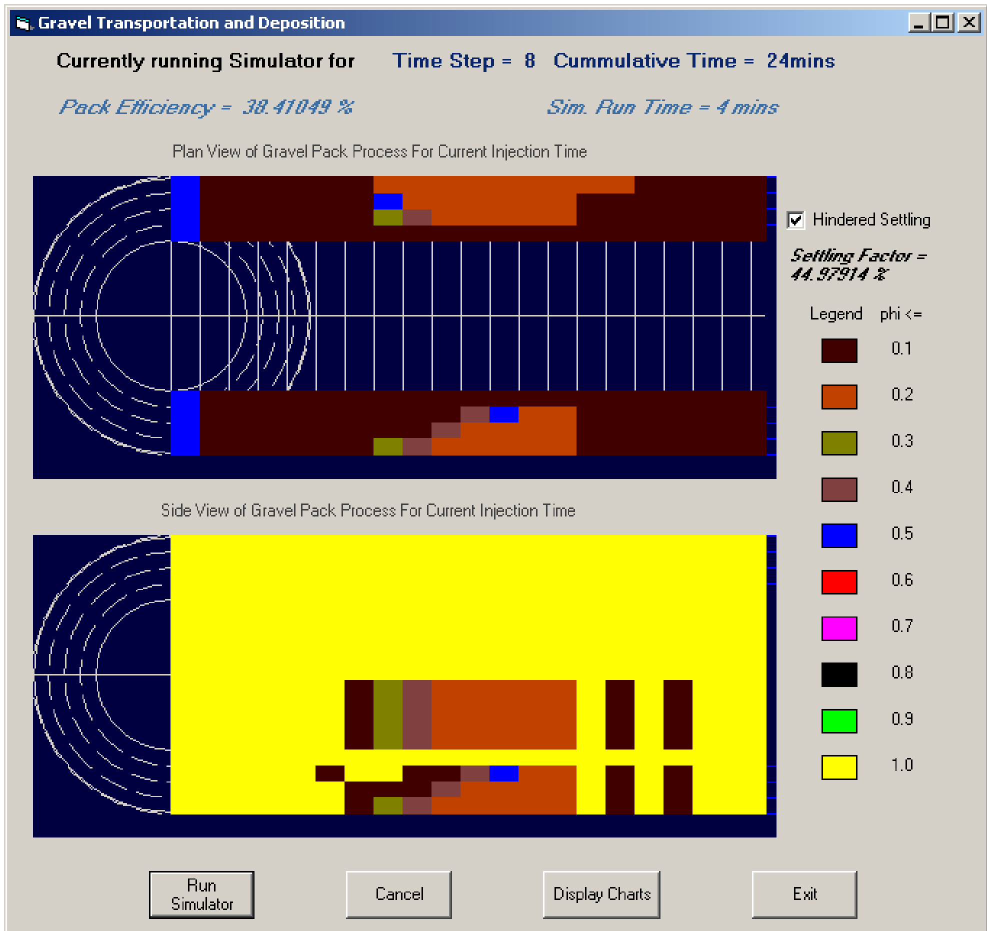


Figure A- 5.44: Example 3 Simulation in Progress for time step = 8

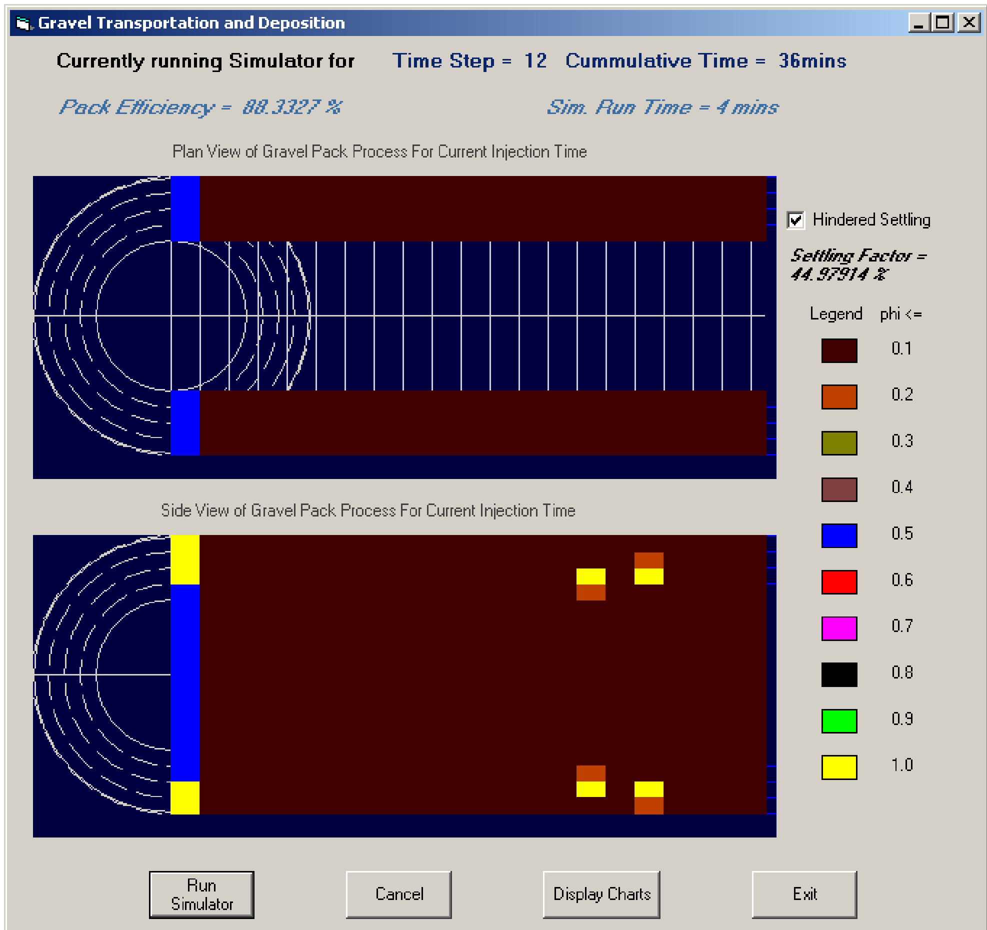


Figure A- 5.45: Example 3 Simulation in Progress for time step = 12

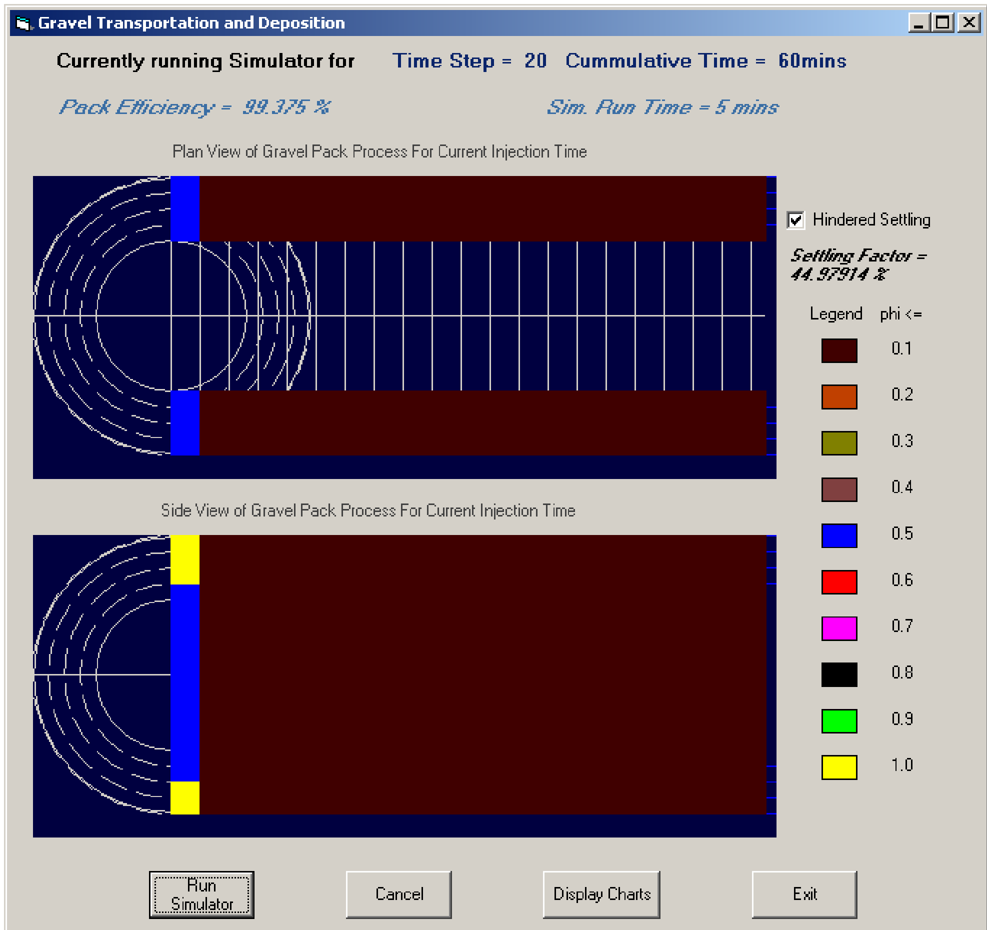


Figure A-5.46: Example 3 Simulation in Progress for time step =20

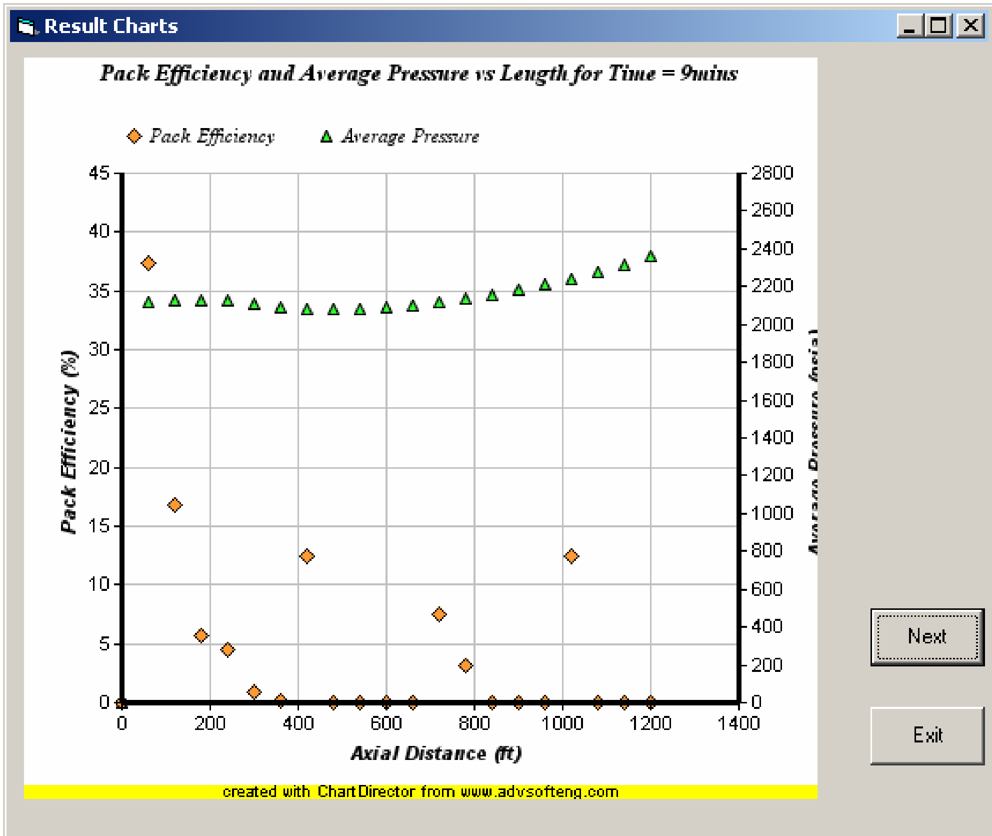


Figure A- 5.47: Example 3 Result Charts Screen at Time = 9min

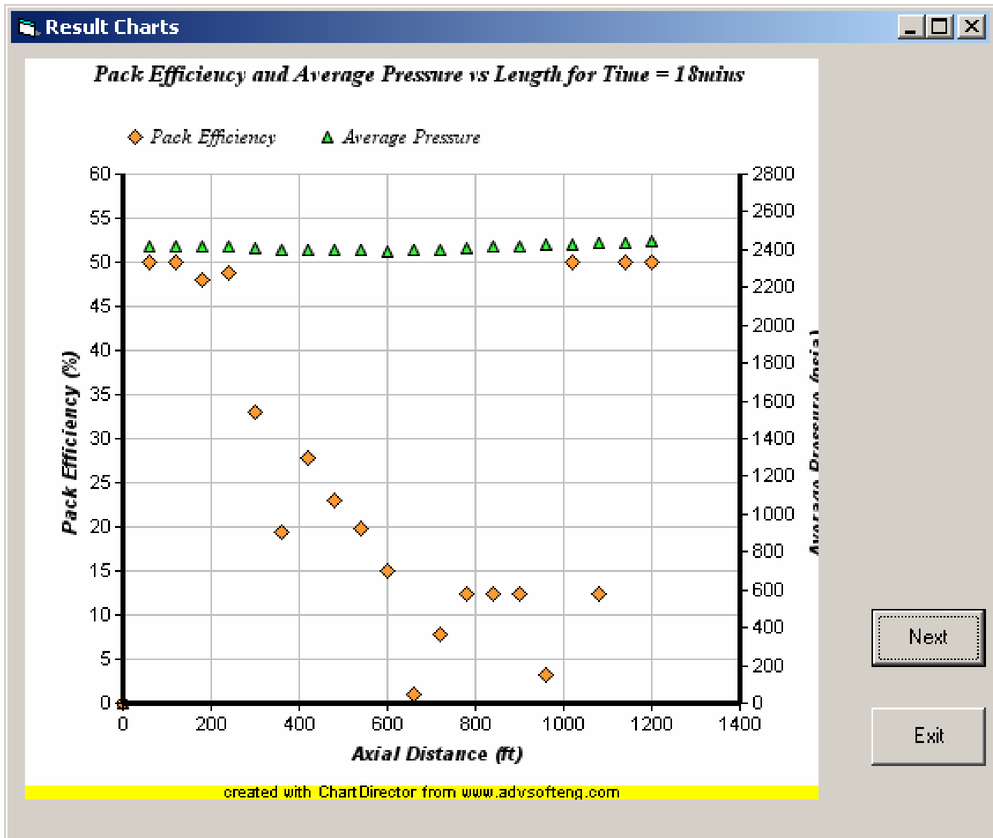


Figure A- 5.48: Example 3 Result Charts Screen at Time = 18min

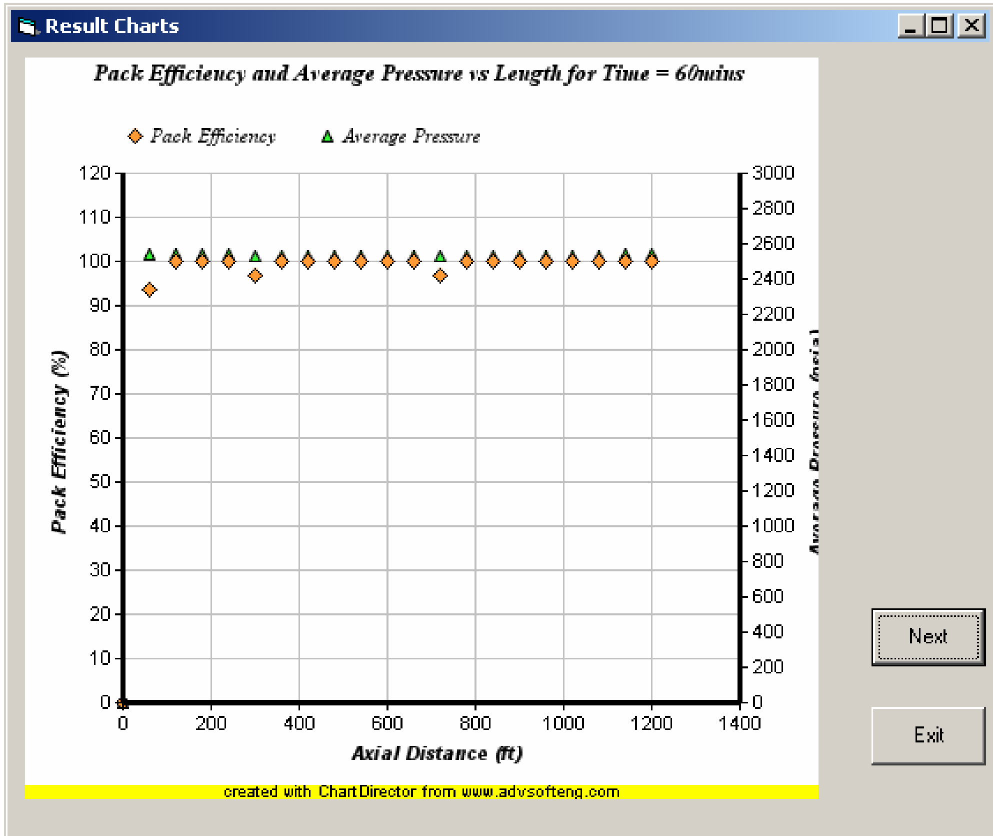


Figure A- 5.49: Example 3 Result Charts Screen at Time = 60min

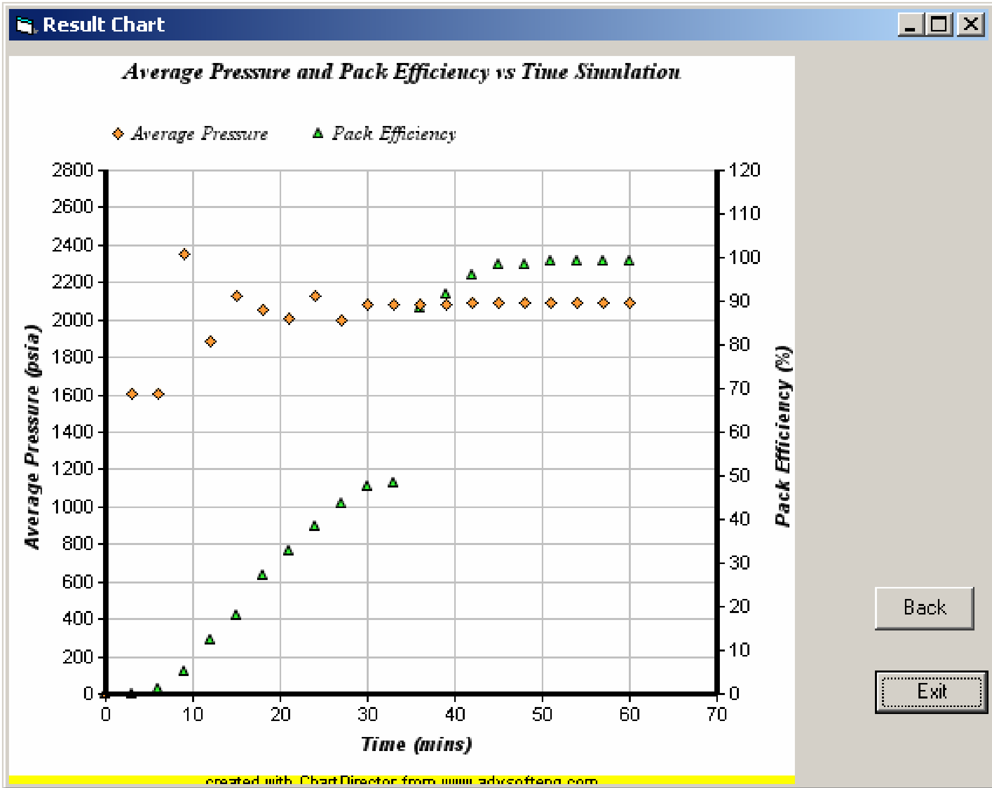


Figure A- 5.50: Example 3 Final Result Charts Screen

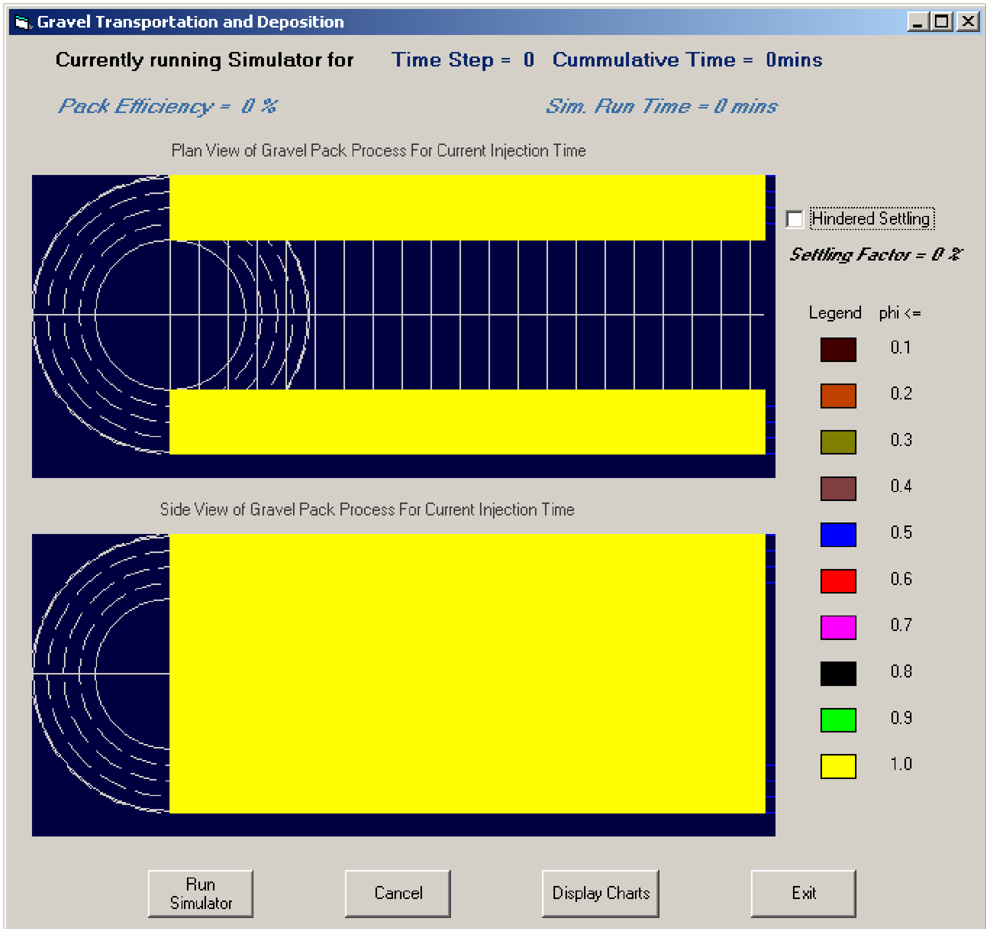


Figure A- 5.51: Example 3 Case 2 Simulation for time step = 0

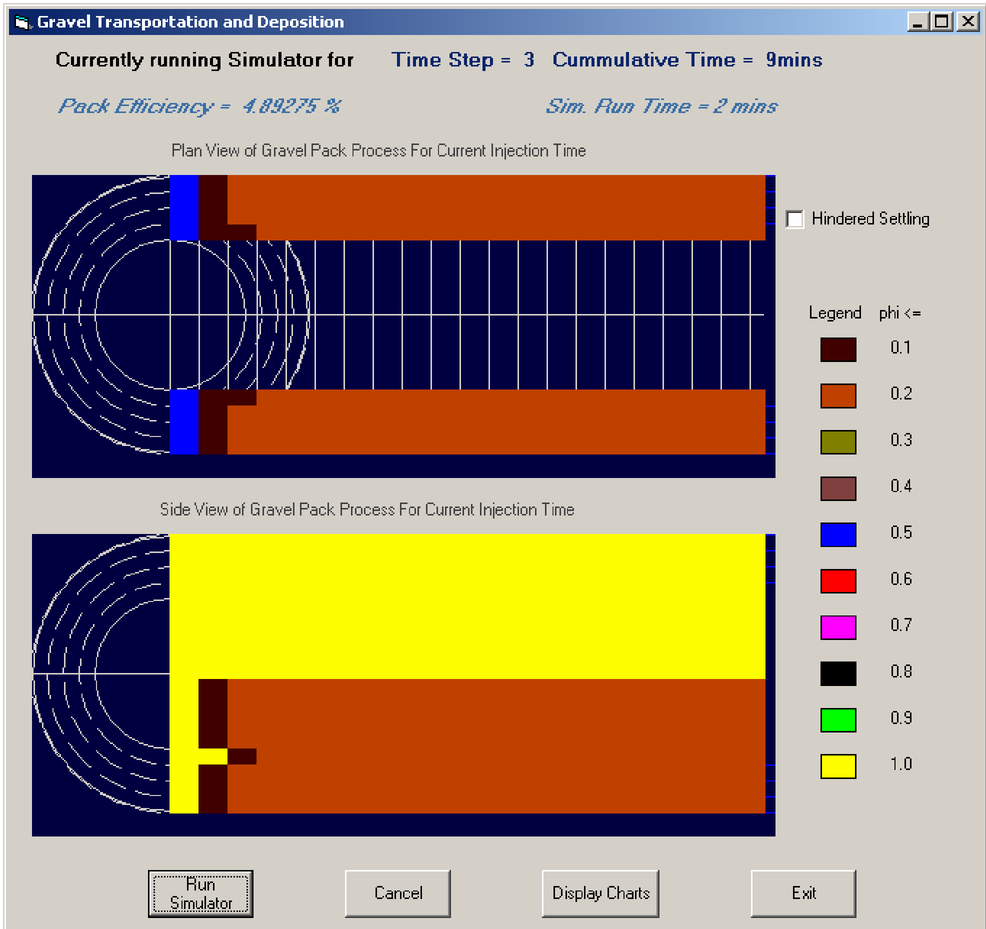


Figure A- 5.52: Example 2 Case 2 Simulation for time step = 3

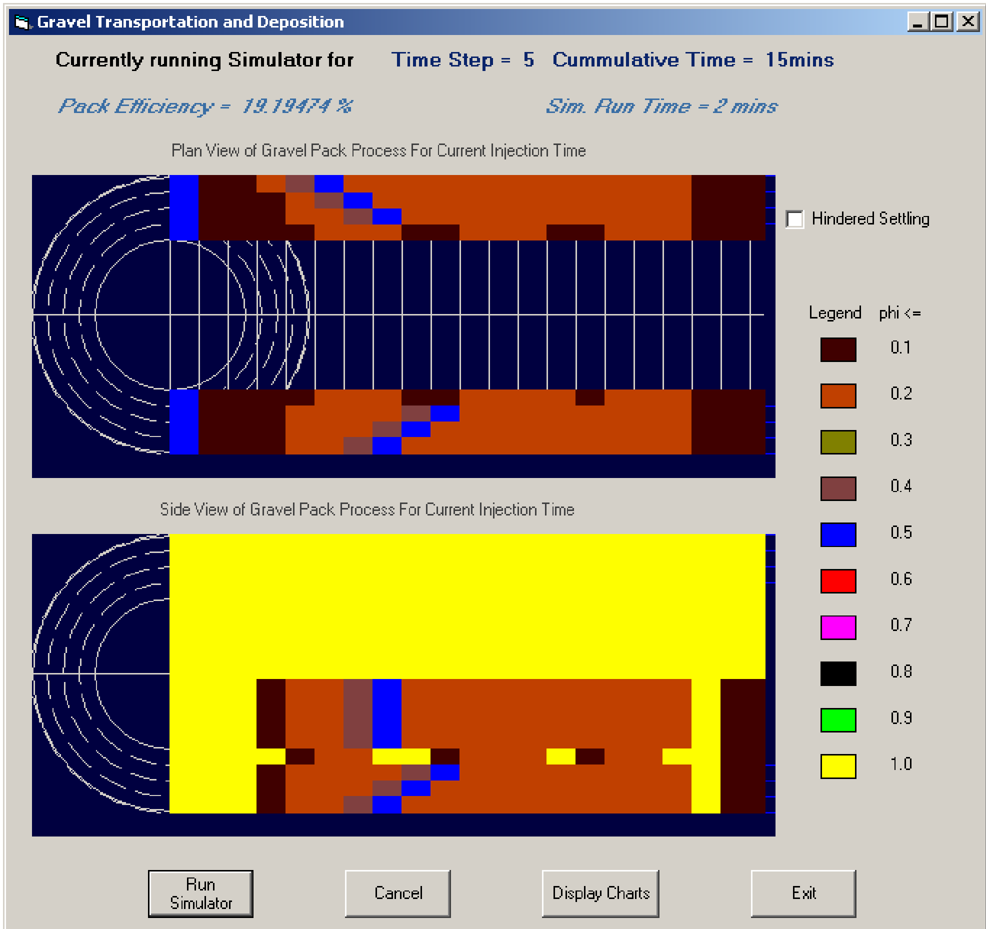


Figure A- 5.53: Example 2 Case 2 Simulation for time step = 15

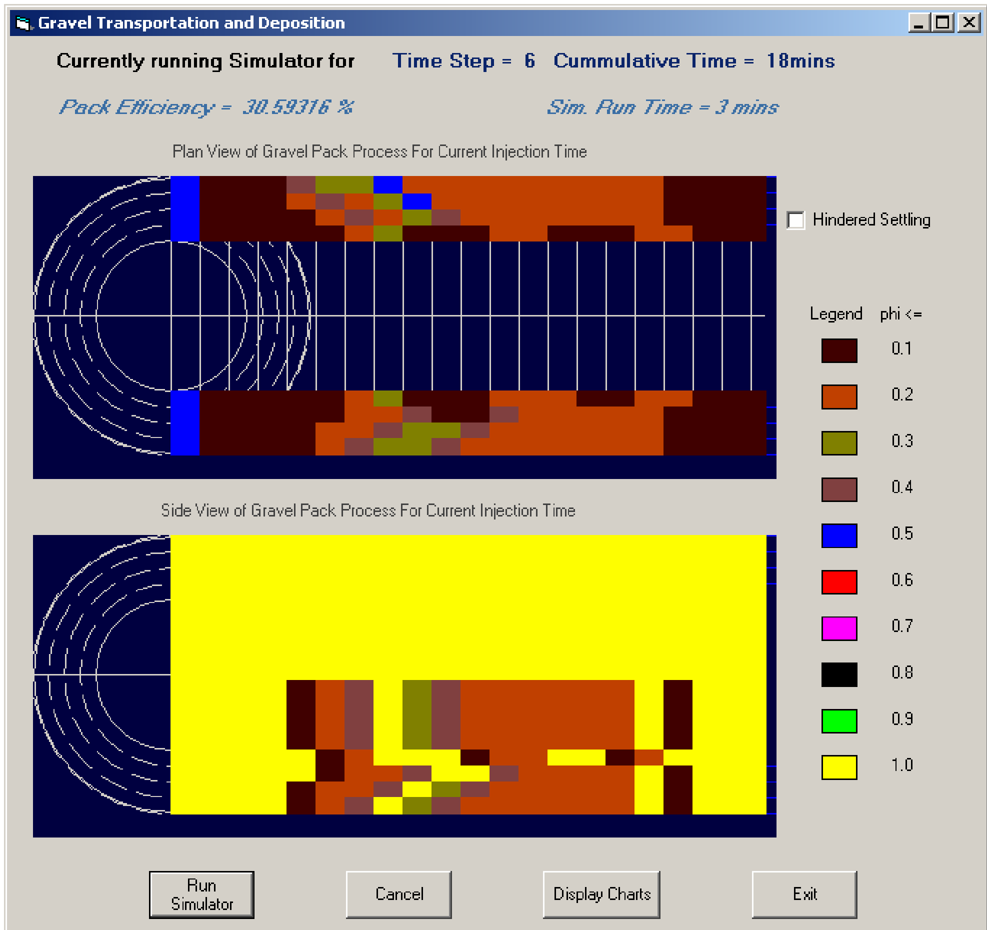


Figure A- 5.54: Example 3 Simulation in Progress for time step = 6

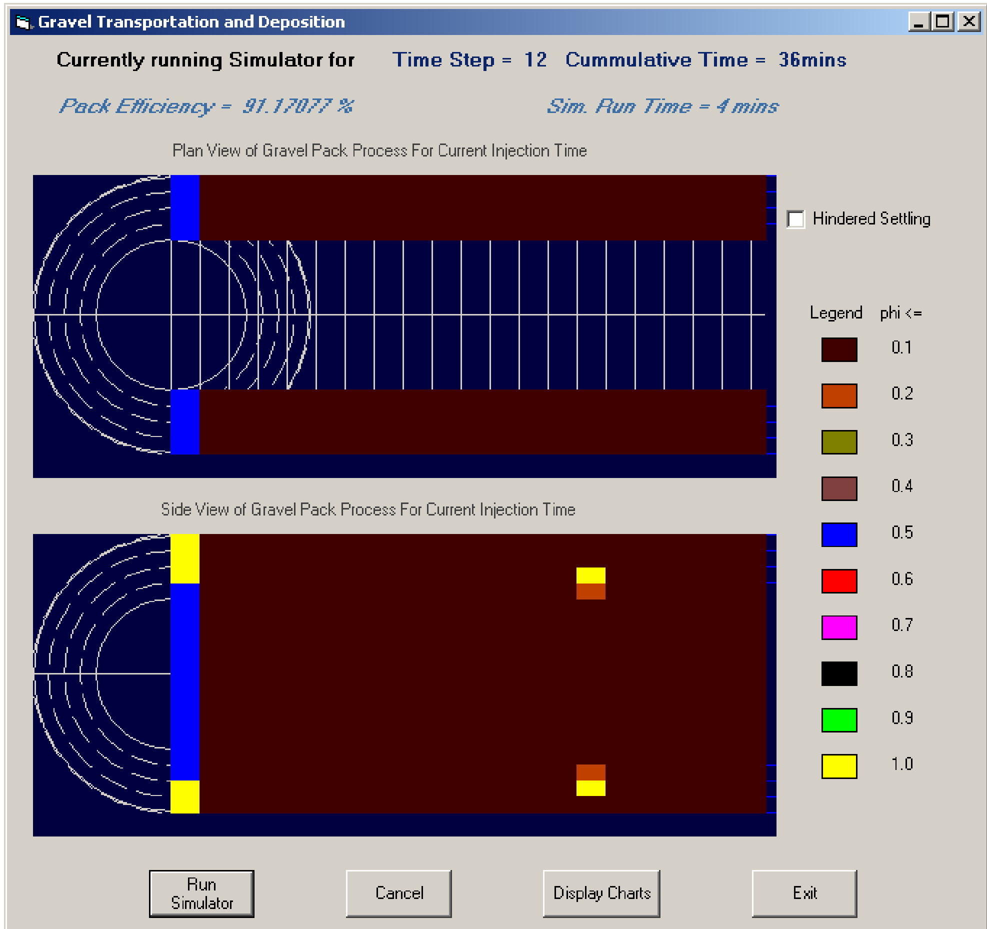


Figure A- 5.55: Example 2 Case 2 Simulation for time step = 12

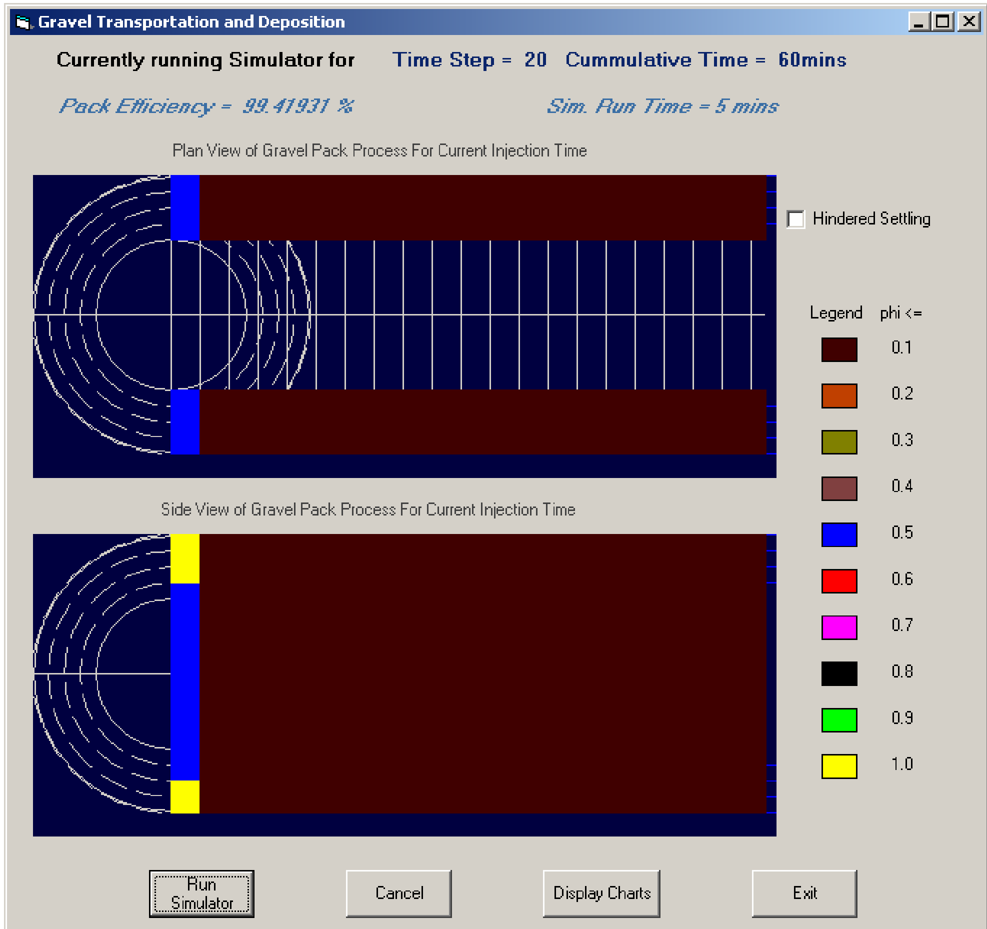


Figure A- 5.56: Example 2 Case 2 Simulation for time step = 20

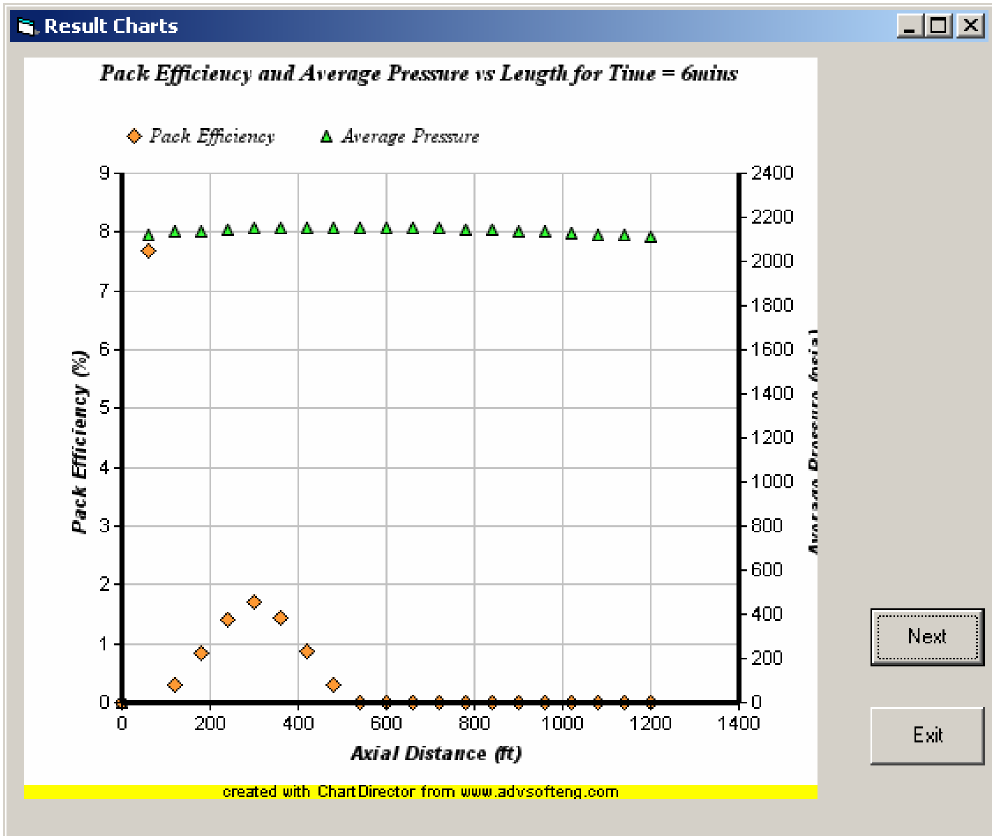


Figure A- 5.57: Example 3 Case 2 Charts Screen at Time = 6min

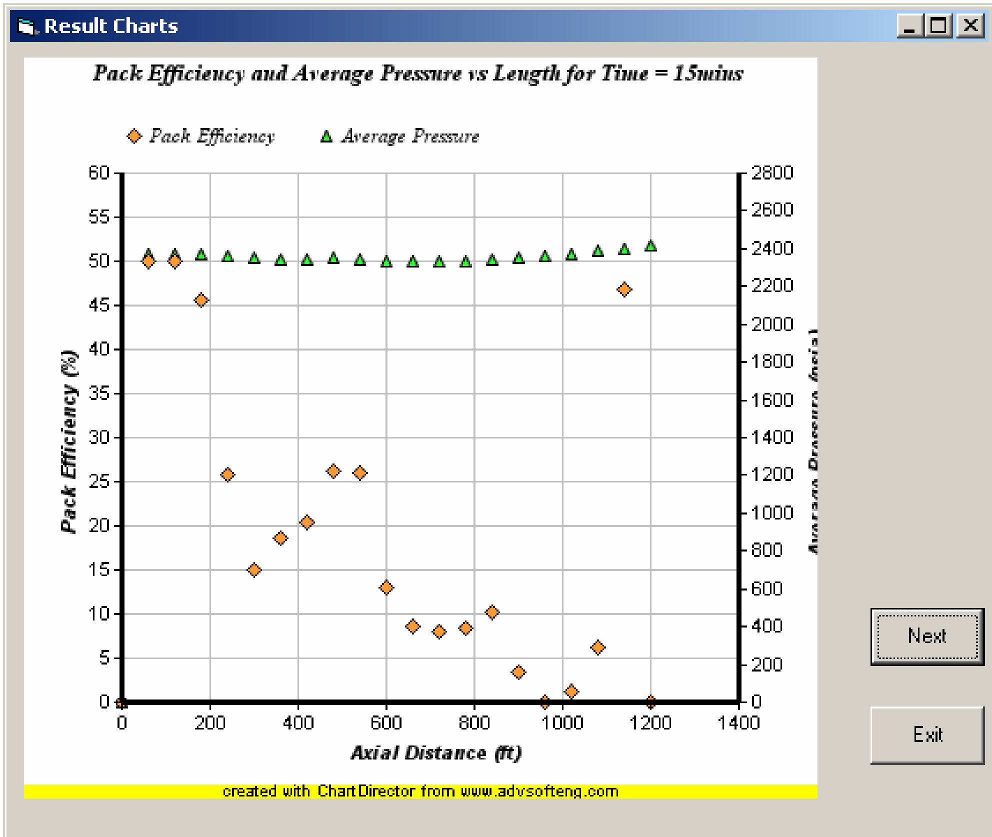


Figure A- 5.58: Example 3 Case 2 Charts Screen at Time = 15min

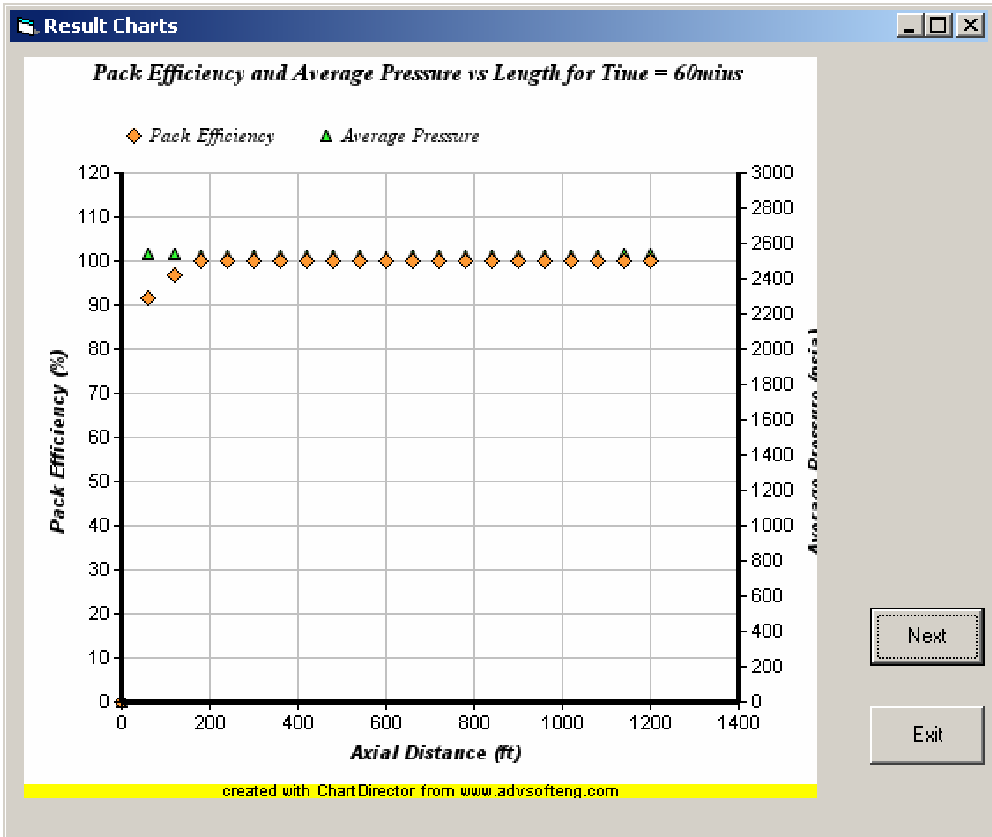


Figure A- 5.59: Example 3 Case 2 Charts Screen at Time = 60min

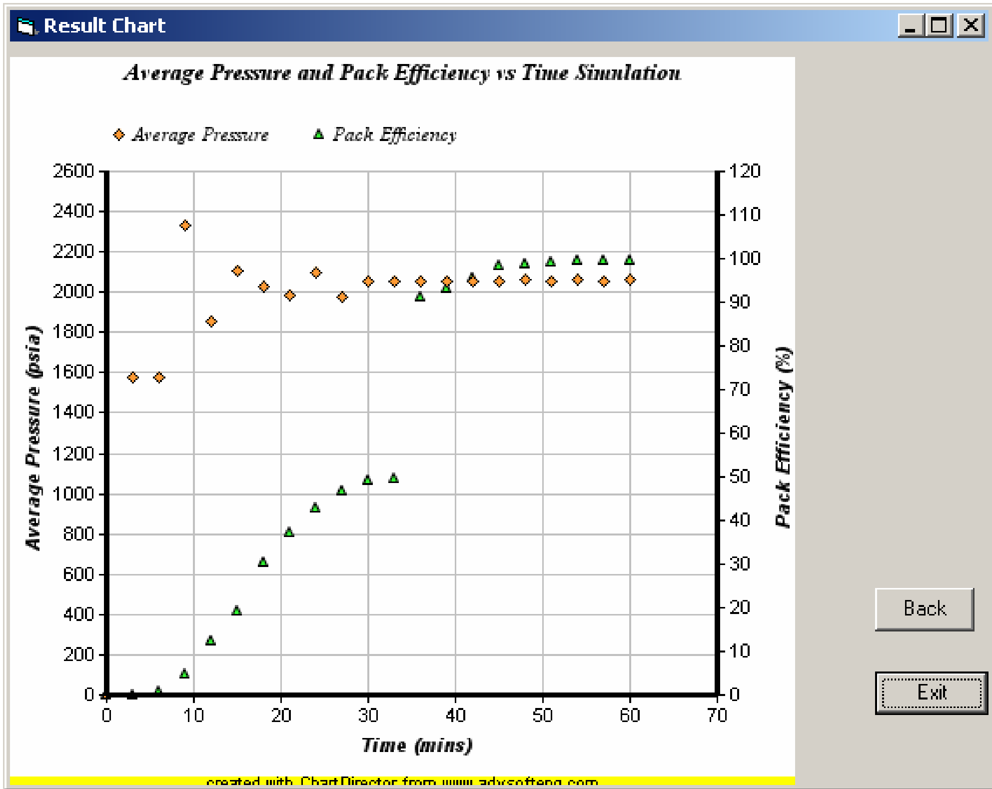


Figure A- 5.60: Example 3 Case 2 Final Result Charts Screen

APPENDIX B: HGPSIM PROGRAM LISTING

```
Public fMainForm As frmMain
' Main function
'Declare the grid dimensions, ic x jc x kc
   Public Const ic = 4
   Public Const jc = 8
   Public Const kc = 20
'Declare the number of time steps
   Public Const nc = 60
'Declare public variables
   Public rho(-2 To nc + 1, -2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public cp(-2 To nc + 1, -2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public vr(-2 To nc + 1, -2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public vz(-2 To nc + 1, -2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public phi(-2 To nc + 1, -2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public kd(-2 To nc + 1, -2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public rad(-2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public Pressure(-2 To nc + 1, -2 To ic + 1, -2 To jc + 1, -2 To kc + 1) As Single
   Public rhoo As Single, cpo As Single, vro As Single,
   Public vzo As Single, Po As Single, phio As Single
   Public A As Single, B As Single, c As Single, D As Single, E As Single
   Public F As Single, G As Single, H As Single, I1 As Single
   Public J1 As Single, dzp As Single
   Public dr, dteta, dz, dt, vis, trad, kratio, MX(8000) As Single
   Public vstvd, vsmd, vslcs, vsohd, pi, gz, gr, gteta As Single
   Public hshd, hshdf, hl, sdia, sdiaf, sl, hscd, ttd, ttl, wpod, wpid As Single
   Public Phir, Kr, Pr, Fg, skin, Co, re, Tr, t, fc, ttotal As Single
   Public cfd, cfv, gd, gdia, gc_ppg, sd, sv, Ts, Q, Ps, cfn, cfk, jobvol As Single
   Public ProjectName, WellName As String, FieldName As String
   Public PreparedBy As String, ClientName As String, Location As String
   Public PrepDate As Date, fmdepth As Single
   Public overburden As Single, startTime As Single, currentTime As Single
   Public iteration, stepn As Integer, condition As Boolean
   Public PackEfficiency As Single, avgPressure(nc) As Single
   Public avgVelocity(nc) As Single
   Public vt As Single, es As Single, avgVflux(nc) As Single
   Public Packeff(nc) As Single, setfactor As Single
 ' Subroutine to freeze the display for s seconds for viewing
   Public Sub wait(s)
   start = Timer   ' Set start time.
   Do While Timer < start + s
     DoEvents   ' Yield to other processes.
   Loop
   finish = Timer   ' Set end time.
   TotalTime = finish - start   ' Calculate total time.
   End Sub
```

```
‘ Subroutine to evaluate the increases in the independent variables usinga modified ‘ ‘
‘  Newtons Method
    Public Sub evaldelta(n, fc)
    Dim i As Integer, j As Integer, k, l, m, n2, L1 As Integer
    Dim funct(8000) As Single, fprime(8000, 8000) As Single, ttemp As Single
    'Dim funct(L1) As Single, fprime(L1, L1) As Single, ttemp As Single
    Dim cvar1 As Integer, cvar2 As Integer, cvar3
    Public klast, khigh, klow, kstep As Integer, diff As Single
    cvar1 = 0
    cvar2 = 0
    cvar3 = 0
     ' do procedure for 10 z step at a time
    klast = kc / 10
    For kstep = 0 To klast - 1
    klow = kstep * 10 + 1
    khigh = (kstep + 1) * 10
    For i = 1 To ic
    For j = 1 To jc
    For k = klow To khigh

‘ Calculate the matrix system AX = F
' where A = fprime(cvar2, cvar3), F = funct(cvar1) and X is the solution matrix
MX(cvar1)
 ' calculate for f1 system
      cvar1 = cvar1 + 1
      funct(cvar1) = -f1(n, i, j, k)
      MX(cvar1) = 0
       cvar2 = cvar2 + 1
       For l = 1 To ic
       For m = 1 To jc
       For n2 = klow To khigh
            cvar3 = cvar3 + 1
            ttemp = rho(n, l, m, n2)
            fprime(cvar2, cvar3) = f1(n, i, j, k)
            diff = fc * rho(n, l, m, n2)
            rho(n, l, m, n2) = (1 - fc) * rho(n, l, m, n2)
            If diff = 0 Then
                fprime(cvar2, cvar3) = 0
            Else
                 fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f1(n, i, j, k)) / diff
            End If
            rho(n, l, m, n2) = ttemp
            cvar3 = cvar3 + 1
            ttemp = vr(n, l, m, n2)
            fprime(cvar2, cvar3) = f1(n, i, j, k)
            diff = (fc * vr(n, l, m, n2))
            vr(n, l, m, n2) = (1 - fc) * vr(n, l, m, n2)
            If diff = 0 Then
                fprime(cvar2, cvar3) = 0
            Else
```

```
            fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f1(n, i, j, k)) / diff
         End If
         vr(n, l, m, n2) = ttemp
         cvar3 = cvar3 + 1
         ttemp = vz(n, l, m, n2)
         fprime(cvar2, cvar3) = f1(n, i, j, k)
         diff = (fc * vz(n, l, m, n2))
         vz(n, l, m, n2) = (1 - fc) * vz(n, l, m, n2)
         If diff = 0 Then
            fprime(cvar2, cvar3) = 0
         Else
            fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f1(n, i, j, k)) / diff
         End If

         vz(n, l, m, n2) = ttemp
         cvar3 = cvar3 + 1
         ttemp = Pressure(n, l, m, n2)
         fprime(cvar2, cvar3) = f1(n, i, j, k)
         diff = (fc * Pressure(n, l, m, n2))
         Pressure(n, l, m, n2) = (1 - fc) * Pressure(n, l, m, n2)
         If diff = 0 Then
            fprime(cvar2, cvar3) = 0
         Else
            fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f1(n, i, j, k)) / diff
         End If
         Pressure(n, l, m, n2) = ttemp
       Next n2
       Next m
       Next l
       cvar3 = 0
  Next k
  Next j
  Next i
' calculate for f2 system
  cvar3 = 0
  For i = 1 To ic
  For j = 1 To jc
  For k = klow To khigh
    cvar1 = cvar1 + 1
     funct(cvar1) = -f2(n, i, j, k)
     MX(cvar1) = 0
     cvar2 = cvar2 + 1
      For l = 1 To ic
      For m = 1 To jc
      For n2 = klow To khigh
            cvar3 = cvar3 + 1
            ttemp = rho(n, l, m, n2)
            fprime(cvar2, cvar3) = f2(n, i, j, k)
            diff = (fc * rho(n, l, m, n2))
            rho(n, l, m, n2) = (1 - fc) * rho(n, l, m, n2)
```

```
            If diff = 0 Then
                fprime(cvar2, cvar3) = 0
            Else
                fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f2(n, i, j, k)) / diff
            End If

            rho(n, l, m, n2) = ttemp
            cvar3 = cvar3 + 1
            ttemp = vr(n, l, m, n2)
            fprime(cvar2, cvar3) = f2(n, i, j, k)
            diff = (fc * vr(n, l, m, n2))
            vr(n, l, m, n2) = (1 - fc) * vr(n, l, m, n2)
            If diff = 0 Then
               fprime(cvar2, cvar3) = 0
            Else
               fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f2(n, i, j, k)) / diff
            End If

            vr(n, l, m, n2) = ttemp
            cvar3 = cvar3 + 1
            ttemp = vz(n, l, m, n2)
            fprime(cvar2, cvar3) = f2(n, i, j, k)
            diff = (fc * vz(n, l, m, n2))
            vz(n, l, m, n2) = (1 - fc) * vz(n, l, m, n2)
            If diff = 0 Then
                fprime(cvar2, cvar3) = 0
            Else
                fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f2(n, i, j, k)) / diff
            End If
            vz(n, l, m, n2) = ttemp
            cvar3 = cvar3 + 1
            ttemp = Pressure(n, l, m, n2)
            fprime(cvar2, cvar3) = f2(n, i, j, k)
            diff = (fc * Pressure(n, l, m, n2))
            Pressure(n, l, m, n2) = (1 - fc) * Pressure(n, l, m, n2)
            If diff = 0 Then
                fprime(cvar2, cvar3) = 0
            Else
                fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f2(n, i, j, k)) / diff
            End If
            Pressure(n, l, m, n2) = ttemp
            Next n2
            Next m
            Next l
      cvar3 = 0
      Next k
      Next j
      Next i
  ' calculate for f3 system
 cvar3 = 0
```

```
For i = 1 To ic
For j = 1 To jc
For k = klow To khigh

cvar1 = cvar1 + 1
funct(cvar1) = -f3(n, i, j, k)
MX(cvar1) = 0
cvar2 = cvar2 + 1
For l = 1 To ic
For m = 1 To jc
For n2 = klow To khigh
cvar3 = cvar3 + 1
ttemp = rho(n, l, m, n2)
fprime(cvar2, cvar3) = f3(n, i, j, k)
diff = (fc * rho(n, l, m, n2))
rho(n, l, m, n2) = (1 - fc) * rho(n, l, m, n2)
If diff = 0 Then
fprime(cvar2, cvar3) = 0
Else
fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f3(n, i, j, k)) / diff
End If
rho(n, l, m, n2) = ttemp
cvar3 = cvar3 + 1
ttemp = vr(n, l, m, n2)
fprime(cvar2, cvar3) = f3(n, i, j, k)
diff = (fc * vr(n, l, m, n2))
vr(n, l, m, n2) = (1 - fc) * vr(n, l, m, n2)
If diff = 0 Then
fprime(cvar2, cvar3) = 0
Else
fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f3(n, i, j, k)) / diff
End If
vr(n, l, m, n2) = ttemp
cvar3 = cvar3 + 1
ttemp = vz(n, l, m, n2)
fprime(cvar2, cvar3) = f3(n, i, j, k)
diff = (fc * vz(n, l, m, n2))
vz(n, l, m, n2) = (1 - fc) * vz(n, l, m, n2)
If diff = 0 Then
fprime(cvar2, cvar3) = 0
Else
fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f3(n, i, j, k)) / diff
End If
vz(n, l, m, n2) = ttemp
cvar3 = cvar3 + 1
ttemp = Pressure(n, l, m, n2)
fprime(cvar2, cvar3) = f3(n, i, j, k)
diff = (fc * Pressure(n, l, m, n2))
Pressure(n, l, m, n2) = (1 - fc) * Pressure(n, l, m, n2)
If diff = 0 Then
```

```
    fprime(cvar2, cvar3) = 0
    Else
    fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f3(n, i, j, k)) / diff
    End If

    Pressure(n, l, m, n2) = ttemp
    Next n2
    Next m
    Next l
    cvar3 = 0
Next k
Next j
Next i
 ' calculate for f4 system
  cvar3 = 0
  For i = 1 To ic
  For j = 1 To jc
  For k = klow To khigh
  cvar1 = cvar1 + 1
  funct(cvar1) = -f4(n, i, j, k)
  MX(cvar1) = 0
  cvar2 = cvar2 + 1
    For l = 1 To ic
    For m = 1 To jc
    For n2 = klow To khigh
    cvar3 = cvar3 + 1
    ttemp = rho(n, l, m, n2)
    fprime(cvar2, cvar3) = f4(n, i, j, k)
    diff = (fc * rho(n, l, m, n2))
    rho(n, l, m, n2) = (1 - fc) * rho(n, l, m, n2)
    If diff = 0 Then
    fprime(cvar2, cvar3) = 0
    Else
    fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f4(n, i, j, k)) / diff
    End If
    rho(n, l, m, n2) = ttemp
    cvar3 = cvar3 + 1
    ttemp = vr(n, l, m, n2)
    fprime(cvar2, cvar3) = f4(n, i, j, k)
    diff = (fc * vr(n, l, m, n2))
    vr(n, l, m, n2) = (1 - fc) * vr(n, l, m, n2)
    If diff = 0 Then
    fprime(cvar2, cvar3) = 0
    Else
    fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f4(n, i, j, k)) / diff
    End If
    vr(n, l, m, n2) = ttemp
    cvar3 = cvar3 + 1
    ttemp = vz(n, l, m, n2)
    fprime(cvar2, cvar3) = f4(n, i, j, k)
```

```
   diff = (fc * vz(n, l, m, n2))
   vz(n, l, m, n2) = (1 - fc) * vz(n, l, m, n2)
   If diff = 0 Then
   fprime(cvar2, cvar3) = 0
   Else
   fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f4(n, i, j, k)) / diff
   End If
   vz(n, l, m, n2) = ttemp
   cvar3 = cvar3 + 1
   ttemp = Pressure(n, l, m, n2)
   fprime(cvar2, cvar3) = f4(n, i, j, k)
   diff = (fc * Pressure(n, l, m, n2))
   Pressure(n, l, m, n2) = (1 - fc) * Pressure(n, l, m, n2)
   If diff = 0 Then
   fprime(cvar2, cvar3) = 0
   Else
   fprime(cvar2, cvar3) = (fprime(cvar2, cvar3) - f4(n, i, j, k)) / diff
   End If
    Pressure(n, l, m, n2) = ttemp
   Next n2
   Next m
   Next l
   cvar3 = 0
 Next k
 Next j
 Next i
 cvar3 = 0
 L1 = 4 * ic * jc * klast
 Call SOLVMAT(fprime, MX, funct, L1)
 Call adddelta(n, klow)
 Call evalPackEfficiency(n)
 Call evalAvgpressure(n)
 Next kstep
 Call evalphi(n)
 Call evalPackEfficiency(n)
 frmgpp.Show
 frmgpp.Refresh
 End Sub

‘ Subroutine to solve the system of matrix MA . MX = MB
     Public Sub SOLVMAT(MA, MX, MB, n)
     Dim outtimer As Timer
     Dim SUM, save, ratio, value As Single, det As Single
     Dim i As Integer, j As Integer, k As Integer, ipvt(8000), ipvtemp As Integer
     Dim nminus1, iplus1, l, kcol, jcol, jrow, tempipvt As Integer
     Dim Space As String, PROMPT As String
     Space = "   "
     ' begin LU decomposition
      det = 1#
     nminus1 = n - 1
```

```
    For i = 1 To n
    ipvt(i) = i
    Next i

    For i = 1 To n - 1
    iplus1 = i + 1
    ipvtemp = i
    For j = iplus1 To n
    If (Abs(MA(ipvtemp, i)) < Abs(MA(j, i))) Then
    ipvtemp = j
    End If
    Next j
    If (ipvtemp <> i) Then
    tempipvt = ipvt(i)
    ipvt(i) = ipvt(ipvtemp)
    For jcol = 1 To n
    save = MA(i, jcol)
    MA(i, jcol) = MA(ipvtemp, jcol)
    MA(ipvtemp, jcol) = save
    Next jcol
    ipvt(ipvtemp) = tempipvt
    det = -det
    End If
' reduce all element below the ith row
    For jrow = iplus1 To n
    If (MA(jrow, i) <> 0) Then
    MA(jrow, i) = MA(jrow, i) / MA(i, i)
    For kcol = iplus1 To n
    MA(jrow, kcol) = MA(jrow, kcol) - MA(jrow, i) * MA(i, kcol)
    Next kcol
    End If
    Next jrow
    Next i
   For i = 1 To n
   MX(i) = MB(ipvt(i))
   Next i
  For irow = 2 To n
   SUM = MX(irow)
   For jcol = 1 To irow - 1
   SUM = SUM - MA(irow, jcol) * MX(jcol)
   Next jcol
   MX(irow) = SUM
   Next irow
   For i = 1 To n
   MB(ipvt(i)) = MX(i)
   Next i
   For i = 1 To n
   MX(i) = MB(i)
   Next i
End Sub
```

```
' Main Form Main Program
   Sub Main()
   Dim fLogin As New frmLogin
   fLogin.Show vbModal
   If Not fLogin.OK Then
       'Login Failed so exit app
       End
   End If
   Unload fLogin
  Set fMainForm = New frmMain
   fMainForm.Show
End Sub

' Subroutine runsim start and directs the simulator computations
Public Sub runsim()
  Dim fc As Single, counter As Integer
   startTime = Timer
   PackEfficiency = 0
   pi = 4 * Atn(1)
   ttotal = 1
   fc = 0.001
   condition = False
   iteration = 0
   dr = (hshd - sdia) / (12 * 2 * ic)
   dteta = 2 * pi / jc
   dz = hl / kc
   dz1 = dz
   dt = ( jobvol / Q  * 60) / nc
   gr = 32
   gteta = 0
   gz = 0
   A = 2 * dz1 * dr
   B = dt * dz1
   c = dr * dt
   D = 2 * (dr ^ 2) * (dteta ^ 2) * (dz1 ^ 2)
   E = dt * dr * (dteta ^ 2) * (dz1 ^ 2)
   F = dt * (dr ^ 2) * (dteta ^ 2) * dz1
   G = 2 * (dteta ^ 2) * (dz1 ^ 2) * dt
   H = 2 * (dr ^ 2) * (dz1 ^ 2) * dt
   I1 = 2 * (dr ^ 2) * (dteta ^ 2) * dt
   J1 = 2 * dt * (dr ^ 2) * (dteta ^ 2) * (dz1 ^ 2)
   vis = sv
    rho = sd
    hshdf = hshd / 12
   sdiaf = sdia / 12
   phio = 1#
   trad = sdia / 24
   kratio = (sdia) / hshd
   dzp = dr * ic / 2
```

```
   Po = Pr + (Q * 5.615 / 60) * 8 * vis * dzp / ((2 * pi) * ((hshdf / 2) ^ 4) * ((1 - kratio ^ 4)
- ((1 - kratio ^ 2) ^ 2 / (Log(1 / kratio)))))
   'vzo = (Po - Pr) * ((hshdf / 2) ^ 2) / (4 * vis * dzp / 2) * (1 - (kratio) ^ 2 + ((1 - kratio ^
2) / (Log(1 / kratio)) * Log(kratio)))
    vzo = (Po - Pr) * ((hshdf / 2) ^ 2) / (4 * vis * dzp / 2) * (1 - ((sdiaf + ic * dr / 2) / hshdf)
^ 2 + ((1 - kratio ^ 2) / (Log(1 / kratio)) * Log(((sdiaf + ic * dr / 2) / hshdf))))
    cpo = (rhoo - cfd) / (gd - cfd)
    vro = 0
' initialize all grid parameter
   trad = sdia / 24
   For n = 1 To nc
   For i = -2 To ic + 1
   For j = -2 To jc + 1
   For k = 1 To kc + 1
   rad(i, j, k) = IIf(trad + (i) * dr < trad, trad, trad + (i) * dr)
   rho(n, i, j, k) = IIf((rhoo - 0.001 * (n) * rhoo) < cfd, (rhoo - 0.001 * (n) * rhoo), cfd)
   vr(n, i, j, k) = IIf(vro - 0.001 * (n) * vro < 0, vro, vro - 0.001 * (n) * vro)
  'vz(n, i, j, k) = (Po - Pr) * ((hshdf / 2) ^ 2) / (4 * vis * dzp / 2) * (1 - (2 * rad(i, j, k) /
hshdf) ^ 2 + ((1 - kratio ^ 2) / (Log(1 / kratio)) * Log((2 * rad(i, j, k) / hshdf))))
   Pressure(n, i, j, k) = IIf(Po - 0.001 * (n) * Po < Po - Pr, Po, Po - 0.001 * (n) * Po)
   phi(n, i, j, k) = 1
   Next k
   Next j
   Next i
   Next n
'initial condition
    n = 0
   For i = -2 To ic + 1
    For j = -2 To jc + 1
    For k = -2 To kc + 1
    rad(i, j, k) = IIf(trad + (i) * dr < trad, trad, trad + (i) * dr)
    rho(n, i, j, k) = rhoo
    vr(n, i, j, k) = vro
    vz(n, i, j, k) = vzo
    Pressure(n, i, j, k) = Po
    phi(n, i, j, k) = 1
    Next k
    Next j
    Next i
' boundary condition
    For n = 0 To nc + 1
    For i = ic To ic + 1
    For j = -2 To jc + 1
    For k = -2 To kc + 1
    If (Pressure(n, i, j, k) > Pr) Then
    vr(n, i, j, k) = Kr / sv * Abs(Pr - Pressure(n, i, j, k)) / rad(i, j, k)
    'vr(n, i, j, k) = 0
    Else
    vr(n, i, j, k) = 0
    End If
```

```
   vz(n, i, j, k) = 0
   Next k
   Next j
   Next i
   Next n

  'initialize kd factor for boundary cells
   For n = 0 To nc
   For i = -2 To 0
   For j = -2 To 0
   For k = -2 To 0
   phi(n, i, j, k) = 1
   kd(n, i, j, k) = 1
   Next k
   Next j
   Next i
   Next n
   For n = 0 To nc
   phi(n, ic + 1, jc + 1, kc + 1) = 1
   kd(n, ic + 1, jc + 1, kc + 1) = 1
   Next n
 'Start simulator calculation
   For stepn = 1 To nc
   While (condition = False) Or iteration < 10
   Call evaldelta(stepn, fc)
   If ttotal > 10 Or ttotal < -10 Then
   fc = fc / iteration
   iteration = iteration + 1
   Call evaldelta(stepn, fc)
   Call evalPackEfficiency(stepn)
   Call evalAvgpressure(stepn)
   End If

   If ttotal >= -1 And ttotal <= 1 Then
   iteration = 0
   condition = False
   GoTo cout2
   End If
   Wend
 Call evalphi(stepn)
Call evalPackEfficiency(stepn)
Call evalAvgpressure(stepn)
frmgpp.Show
frmgpp.Refresh
'check if annulus is blocked
For k = 1 To kc
counter = 0
For j = 1 To jc
For i = 1 To ic
If phi(stepn, i, j, k) = 0 Then
```

```
counter = counter + 1
End If
Next i
Next j
If (counter >= ic * jc) Then
counter = 0
GoTo cout
End If
Next k
Next stepn
cout:

For i = 1 To ic
For j = 1 To jc
For k = 1 To kc
phi(nc + 1, i, j, k) = phi(stepn, i, j, k)
Next k
Next j
Next i

If (stepn > nc) Then
stepn = nc
End If
Call evalPackEfficiency(stepn)
Call evalAvgpressure(stepn)
frmgpp.Show
frmgpp.Refresh
End Sub
‘ Subroutine to estimate pack efficiency for each time step
Public Sub evalPackEfficiency(n)
Dim dphi As Single, ccount As Single
ccount = 0
dphi = 0
For i = 1 To ic
For j = 1 To jc
For k = 1 To kc
dphi = dphi + phi(n, i, j, k)
ccount = ccount + 1
Next k
Next j
Next i
PackEfficiency = (1 - dphi / ccount) * 100
Packeff(n) = PackEfficiency
End Sub
‘ Subroutine to estimate average grid pressure for each time step
Public Sub evalAvgpressure(n)
Dim dphi1 As Single, dphi2, dphi3 As Single, ccount As Single
ccount = 0
dphi1 = 0
dphi2 = 0
```

```
dphi3 = 0
For i = 1 To ic
For j = 1 To jc
For k = 1 To kc
ccount = ccount + 1
dphi2 = dphi2 + dr * dz * (rad(i, j, k) + 0.5 * dr) * dteta
dphi1 = dphi1 + Pressure(n, i, j, k) * dr * dz * (rad(i, j, k) + 0.5 * dr) * dteta
If (Pressure(n, i, j, k) > dphi3) Then
dphi3 = Pressure(n, i, j, k)
End If
Next k
Next j
Next i
avgPressure(n) = dphi1 / dphi2
avgPressure(n) = avgPressure(n) + 2 * Pfriction - 0.0052 * vstvd * rhoo
End Sub

'Subroutine to estimate available space for deposition for each grid
Public Sub evalphi(n)
Dim dphi As Single
For i = 1 To ic
For j = 1 To jc
For k = 1 To kc
cp(n, i, j, k) = (rho(n, i, j, k) - cfd) / (gd - cfd)

' set kd values
If ((dteta * j > pi) And ((n * dt * Q * 5.615 / 60) < (pi * hl * ((hshdf / 2) ^ 2 - (sdiaf / 2) ^
2)))) Then
kd(n, i, j, k) = 0
Else
kd(n, i, j, k) = 1 / (dt * Sqr((vro ^ 2) + (vzo ^ 2)) * cpo)
End If
Next k
Next j
Next i

For i = 1 To ic
For j = 1 To jc
For k = 1 To kc
dphi = IIf((dt * kd(n, i, j, k) * Sqr((vr(n, i, j, k) ^ 2) + (vz(n, i, j, k) ^ 2)) * (phi(n - 1, i, j, k) ^
(2 / 3)) * cp(n, i, j, k)) > 1, 1, dt * kd(n, i, j, k) * Sqr((vr(n, i, j, k) ^ 2) + (vz(n, i, j, k) ^ 2)) *
(phi(n - 1, i, j, k) ^ (2 / 3)) * cp(n, i, j, k))
dphi = IIf(dphi < 0, 0, dphi)
phi(n, i, j, k) = IIf((-dphi + phi(n - 1, i, j, k)) < 0, 0, (-dphi + phi(n - 1, i, j, k)))
Next k
Next j
Next i
End Sub

' Subroutine to ensure integrity of calculated deltas
```

```
Public Sub evalcond(n, klow)
Dim cvar1 As Single, ttotal As Single
cvar1 = 0
ttotal = 0
For i = 1 To ic
For j = 1 To jc
For k = klow To klow + 9
cvar1 = cvar1 + 1
If Abs(MX(cvar1)) > 2 * rho(n, i, j, k) Then
MX(cvar1) = 0
End If
cvar1 = cvar1 + 1
'If Abs(MX(cvar1)) > 15 * vr(n, i, j, k) Then
If Abs(MX(cvar1)) > 2 * vz(n, i, j, k) Then
MX(cvar1) = 0
End If
cvar1 = cvar1 + 1
If Abs(MX(cvar1)) > 2 * vz(n, i, j, k) Then
MX(cvar1) = 0
End If
cvar1 = cvar1 + 1
If Abs(MX(cvar1)) > 2 * Pressure(n, i, j, k) Then
MX(cvar1) = 0
End If
Next k
Next j
Next i
cvar1 = 0
End Sub

'Subroutine to generate output file to text file
Public Sub saveall()
Dim sFile As String, t As Integer
   sFile = App.Path & "\output\"
   sFile1 = sFile & ProjectName & "hgpresult.txt"
   Dim fso, txtfile
   Set fso = CreateObject("Scripting.FileSystemObject")
   Set txtfile = fso.CreateTextFile(sFile1, True)
   ' Save Project information
      linetowrite = "Horizontal Well Simulation Run for: " & ProjectName & " Run on :  "
& Now()
   txtfile.WriteLine (linetowrite)
   linetowrite = "Final Pack Efficiency = " & PackEfficiency & " %"
   txtfile.WriteLine (linetowrite)
    linetowrite = "Total Simulator Run Time = " & (currentTime - startTime) / 60 & "
mins"
   txtfile.WriteLine (linetowrite)
    If frmgpp.chkHindered = 1 Then
   linetowrite = "The Settling Factor = " & setfactor & " %"
   txtfile.WriteLine (linetowrite)
```

```
    Else
    linetowrite = "The Settling Factor =  No settling Effect"
    txtfile.WriteLine (linetowrite)
    End If
    Linetowrite = "====================================================="
    txtfile.WriteLine (linetowrite)
    linetowrite = "Time" & Chr(9) & "Surface Injection Pressure" & Chr(9) & "Pack
Efficiency "
    txtfile.WriteLine (linetowrite)
    linetowrite = "(mins)" & Chr(9) & "(Psia)" & Chr(9) & "(%)"
    txtfile.WriteLine (linetowrite)

    For t% = 0 To stepn
    linetowrite = t * dt / 60 & Chr(9) & avgPressure(t) & Chr(9) & Packeff(t)
    txtfile.WriteLine (linetowrite)
    Next t

    linetowrite = "===================================================="
    txtfile.WriteLine (linetowrite)
    linetowrite = "Time" & Chr(9) & "Grid [i,j,k]" & Chr(9) & "Average Pressure" & Chr(9)
& _
    "Density" & Chr(9) & "vz" & Chr(9) & "vr" & Chr(9) & "phi"
    txtfile.WriteLine (linetowrite)
   linetowrite = "(mins)" & Chr(9) & "Grid [i,j,k]" & "(Psia)" & Chr(9) & _
    "(lbs/gal)" & Chr(9) & "ft/sec" & Chr(9) & "ft/sec" & Chr(9) & "fraction"
    txtfile.WriteLine (linetowrite)
For t% = 0 To stepn
    For i = 1 To ic
    For j = 1 To jc
    For k = 1 To kc
    linetowrite = t * dt / 60 & Chr(9) & "[" & i & "," & j & "," & k & "]" & Chr(9) &
Pressure(t, i, j, k) & Chr(9) & _
    rho(t, i, j, k) & Chr(9) & vz(t, i, j, k) & Chr(9) & vr(t, i, j, k) & Chr(9) & phi(t, i, j, k)
    txtfile.WriteLine (linetowrite)
    Next k
    Next j
    Next i
    Next t
    txtfile.Close
  End Sub
'Subroutine to add calculated change in independent variables to the previous values
Public Sub adddelta(n, klow)
Dim cvar1 As Single, ttotal As Single, nt As Integer
cvar1 = 0
ttotal = 0
For i = 1 To ic
For j = 1 To jc
For k = klow To klow + 9
cvar1 = cvar1 + 1
rho(n, i, j, k) = rho(n, i, j, k) + MX(cvar1)
```

```
ttotal = ttotal + MX(cvar1)
cvar1 = cvar1 + 1
vr(n, i, j, k) = vr(n, i, j, k) + MX(cvar1)
ttotal = ttotal + MX(cvar1)
cvar1 = cvar1 + 1
vz(n, i, j, k) = vz(n, i, j, k) + MX(cvar1)
ttotal = ttotal + MX(cvar1)
cvar1 = cvar1 + 1
Pressure(n, i, j, k) = Pressure(n, i, j, k) + MX(cvar1)
ttotal = ttotal + MX(cvar1)
Next k
Next j
Next i
ttotal = ttotal / cvar1
End Sub
‘ Function to estimate the first partial differential equation
Public Function f1(n, i, j, k)
f1 = A * rho(n + 1, i, j, k) - A * rho(n, i, j, k) + B * vr(n, i, j, k) * (rho(n, i + 1, j, k) - rho(n, i
- 1, j, k))
f1 = f1 + c * vz(n, i, j, k) * (rho(n, i, j, k + 1) - rho(n, i, j, k - 1))
f1 = f1 + rho(n, i, j, k) * (B * (vr(n, i + 1, j, k) - vr(n, i - 1, j, k)) + c * (vz(n, i, j, k + 1) -
vz(n, i, j, k - 1)))
End Function

‘ Function to estimate the settiling effect Es
Public Function evalEs(vm, vt, cs, tol)
Dim tempval As Double, s As Double, tdiff As Double
ccount = 0
tempval = 0
es = cs
tdiff = tol + 1
While tdiff > tol Or ccount < 1001
s = vt / (1 - es)
tempval = (Fes(es * 1.1, vm, vt, cs) - Fes(es, vm, vt, cs)) / (es * 0.1)
tempval = es - Fes(es, vm, vt, cs) / tempval
tdiff = Abs(es - tempval) / es
es = tempval
ccount = ccount + 1
Wend
If ccount >= 1000 And tdiff > tol Then
es = cs
End If
End Function

Public Function Fes(es, vm, vt, cs)
Dim s As Double
s = vt / (1 - es)
Fes = es + ((vm + s) / (2 * s)) - ((((vm + s) / (2 * s)) ^ 2 + (vm * cs / s)) ^ 0.5)
End Function
```

```
‘ Function to estimate the second  partial differential equation
Public Function f2(n, i, j, k)
Dim jteta As Double
jteta = dteta * j
f2 = rho(n, i, j, k) * (D * (vr(n + 1, i, j, k) - vr(n, i, j, k)) + E * vr(n, i, j, k) * (vr(n, i + 1, j, k)
- vr(n, i - 1, j, k)) + F * vz(n, i, j, k) * (vr(n, i, j, k + 1) - vr(n, i, j, k - 1)))
f2 = f2 + E * (Pressure(n, i + 1, j, k) - Pressure(n, i - 1, j, k))
f2 = f2 - vis * (G * (vr(n, i + 1, j, k) - 2 * vr(n, i, j, k) + vr(n, i - 1, j, k)))
f2 = f2 - vis * (H / rad(i, j, k) ^ 2) * (vr(n, i, j + 1, k) - 2 * vr(n, i, j, k) + vr(n, i, j - 1, k))
f2 = f2 - vis * (I1 * (vr(n, i, j, k + 1) - 2 * vr(n, i, j, k) + vr(n, i, j, k - 1)))
f2 = f2 - J1 * rho(n, i, j, k) * gr * Sin(jteta)
End Function

‘ Function to estimate the third partial differential equation
Public Function f3(n, i, j, k)
f3 = Pressure(n, i, j + 1, k) - Pressure(n, i, j - 1, k) - (2 * vis / (rad(i, j, k)) * (vr(n, i, j + 1,
k) - vr(n, i, j - 1, k)))
f3 = f3 - 2 * dteta * rad(i, j, k) * rho(n, i, j, k) * gteta
End Function

‘ Function to estimate the fourth partial differential equation
Public Function f4(n, i, j, k)
f4 = rho(n, i, j, k) * (D * (vz(n + 1, i, j, k) - vz(n, i, j, k)) + E * vr(n, i, j, k) * (vz(n, i + 1, j,
k) - vz(n, i - 1, j, k)) + F * vz(n, i, j, k) * (vz(n, i, j, k + 1) - vz(n, i, j, k - 1)))

f4 = f4 + F * (Pressure(n, i, j, k + 1) - Pressure(n, i, j, k - 1))
f4 = f4 - vis * (G * (vz(n, i + 1, j, k) - 2 * vz(n, i, j, k) + vz(n, i - 1, j, k)))
f4 = f4 - vis * (H / rad(i, j, k) ^ 2) * (vz(n, i, j + 1, k) - 2 * vz(n, i, j, k) + vz(n, i, j - 1, k))
f4 = f4 - vis * (I1 * (vz(n, i, j, k + 1) - 2 * vz(n, i, j, k) + vz(n, i, j, k - 1)))
f4 = f4 - J1 * rho(n, i, j, k) * gz
End Function
```

```
‘Module ModGraph for rendering the simulator wellbore status
'Setpixel api to draw dots
Private Declare Function SetPixelV Lib "gdi32" (ByVal hdc As Long, ByVal X As Long,
ByVal Y As Long, ByVal crColor As Long) As Long
Private Declare Function LineTo Lib "gdi32" (ByVal hdc As Long, ByVal X As Long,
ByVal Y As Long) As Long
Private Declare Function MoveToEx Lib "gdi32" (ByVal hdc As Long, ByVal X As
Long, ByVal Y As Long, lpPoint As POINTAPI) As Long
Private Type POINTAPI
    X As Long
    Y As Long
End Type
'Vars
Private intXP As Integer ' xpoint used for line draw
Private intYP As Integer ' ypoint used for line draw
Private intGW As Integer 'graw width
Private intGH As Integer 'graph height
```

```
Private intCenter As Integer 'hole center
Private intOHrad As Integer 'hole center
Private intSCrad As Integer 'hole center
Private intGraphArray() As Integer 'array to hold graph values
Private mintDrawIndex As Integer 'draw style index, 0,1,2, dots, bars, line
Private mintValue As Integer 'graph value to plot
Private mintTimerInterval As Integer 'timer interval of the graph
Private mintStepr As Integer 'every r to point to grid
Private mintStepteta As Integer 'every teta to point to grid
Private mintStepz As Integer 'every z point to grid
'objects
Private tmrGraphTimer As Timer 'timer of the graph
Private pGraphBox As PictureBox 'picture box to draw the graph on
Option Explicit
Public Sub InitGraph(tmr As Timer, picBox As PictureBox, tmrInterval As Integer,
intStepz, intStepteta, intstepr, hcenter, ohrad, scrad)
'Init the graph, with graph width and height.
'Set the module global picture box and timer.
'Redim the grapharray which holds the values of the graph, to width of pgraph picture
box.
'Set the timer interval here(or not), which can be changed in real time if the users
wishes.
   Dim intI As Integer 'index of graph
   Set pGraphBox = picBox
   Set tmrGraphTimer = tmr
   intGW = pGraphBox.Width / 1.25 'graph width max
   intGH = pGraphBox.Height / 2.2 'graph height max
  ' intGW = pGraphBox.Width  'graph width max
  ' intGH = pGraphBox.Height 'graph height max

   intCenter = hcenter ' hole center
   intOHrad = ohrad ' hole rad
   intSCrad = scrad ' screen rad

   Call SetrGridStep(intstepr)
   Call SettetaGridStep(intStepteta)
   Call SetzGridStep(intStepz)
   Call GridGraph
'    ReDim intGraphArray(intGW) 'resize the array
   Call FillGraphArray1
   'mintTimerInterval = tmrInterval
   'tmrGraphTimer.Interval = tmrInterval
   'tmrGraphTimer.Enabled = True 'start the timer
End Sub
Public Sub UpdateGraph(tmr As Timer, picBox As PictureBox, tmrInterval As Integer,
intStepz, intStepteta, intstepr, hcenter, ohrad, scrad)
'Init the graph, with graph width and height.
'Set the module global picture box and timer.
'Redim the grapharray which holds the values of the graph, to width of pgraph picture
box.
```

```
'Set the timer interval here(or not), which can be changed in real time if the users wishes.
    Dim intI As Integer 'index of graph
    Set pGraphBox = picBox
    Set tmrGraphTimer = tmr
    intGW = pGraphBox.Width / 1.25 'graph width max
    intGH = pGraphBox.Height / 2.2 'graph height max
   ' intGW = pGraphBox.Width  'graph width max
   ' intGH = pGraphBox.Height 'graph height max
    intCenter = hcenter ' hole center
    intOHrad = ohrad ' hole rad
    intSCrad = scrad ' screen rad
    pGraphBox.Cls 'clear old picture
    Call SetrGridStep(intstepr)
    Call SettetaGridStep(intStepteta)
    Call SetzGridStep(intStepz)
    Call GridGraph
 '   ReDim intGraphArray(intGW) 'resize the array
    mintTimerInterval = tmrInterval
    'tmrGraphTimer.Interval = tmrInterval
    'tmrGraphTimer.Enabled = True 'start the timer
    Call FillGraphArray1
    Call FillGraphArray2
End Sub
Public Sub UpdateGraph3(tmr As Timer, picBox As PictureBox, tmrInterval As Integer, intStepz, intStepteta, intstepr, hcenter, ohrad, scrad)
'Init the graph, with graph width and height.
'Set the module global picture box and timer.
'Redim the grapharray which holds the values of the graph, to width of pgraph picture box.
'Set the timer interval here(or not), which can be changed in real time if the users wishes.
    Dim intI As Integer 'index of graph
    Set pGraphBox = picBox
    Set tmrGraphTimer = tmr
    intGW = pGraphBox.Width / 1.25 'graph width max
    intGH = pGraphBox.Height / 2.2 'graph height max
   ' intGW = pGraphBox.Width  'graph width max
   ' intGH = pGraphBox.Height 'graph height max

    intCenter = hcenter ' hole center
    intOHrad = ohrad ' hole rad
    intSCrad = scrad ' screen rad
    pGraphBox.Cls 'clear old picture
    Call SetrGridStep(intstepr)
    Call SettetaGridStep(intStepteta)
    Call SetzGridStep(intStepz)
    Call GridGraph
 '   ReDim intGraphArray(intGW) 'resize the array
    mintTimerInterval = tmrInterval
```

```
   'tmrGraphTimer.Interval = tmrInterval
   'tmrGraphTimer.Enabled = True 'start the timer
   Call FillGraphArray1
   Call FillGraphArray3
End Sub

Private Sub GridGraph()
'Draw a grid on the picture box
   Dim intx As Integer
   Dim inty As Integer
   Dim Point As POINTAPI
   pGraphBox.ForeColor = &H8000000F
   'pGraphBox.ForeColor = vbRed
   pGraphBox.FillStyle = 1
   pGraphBox.DrawStyle = 0
   pGraphBox.Cls 'clear old picture
   'If GetXGridStep = 0 Or GetYGridStep = 0 Then Exit Sub 'break out if zero
   If GetzGridStep = 0 Then Call SetzGridStep(dz)
   For intx = intOHrad To intGW + intOHrad Step GetzGridStep 'cycle x's steping each
time
         'pGraphBox.Line (intx, inty)-(intx, intGH), vbRed 'draw x lines
         'pGraphBox.Line (intx, inty)-(intGW, inty), vbRed 'draw y lines

         Point.X = intx: Point.Y = intCenter - intOHrad 'Set the start-point's coordinates
         MoveToEx pGraphBox.hdc, intx, intCenter - intOHrad, Point 'Move the active
point
         LineTo pGraphBox.hdc, intx, intCenter + intOHrad 'Draw a line from the active
point to the given point

   Next intx

  ' Draw empty ellipse for open hole and screen.
       pGraphBox.FillStyle = 1
       pGraphBox.DrawStyle = 0
       pGraphBox.Circle (intOHrad, intCenter), intOHrad, , , , 1
       pGraphBox.Circle (intOHrad, intCenter), intSCrad, , , , 1

  If GetrGridStep = 0 Then Call SetrGridStep(dr)

   For intx = 0 To ic  'cycle r's steping each time
       'Point.X = 0: Point.Y = intCenter

       ' Draw empty ellipse for grids.
       pGraphBox.FillStyle = 1
       pGraphBox.DrawStyle = 1
       pGraphBox.Circle (intOHrad, intCenter), intSCrad + Int(intx * GetrGridStep), , , ,
1
       pGraphBox.FillStyle = 1
       pGraphBox.DrawStyle = 0
       pGraphBox.Line  (intOHrad,  intCenter  +  intSCrad  +  Int(intx  *  GetrGridStep))-
```

```
(intGW + intOHrad, intCenter + intSCrad + Int(intx * GetrGridStep))
      pGraphBox.Line (intOHrad, intCenter - intSCrad - Int(intx * GetrGridStep))-
(intGW + intOHrad, intCenter - intSCrad - Int(intx * GetrGridStep))
   Next intx

   ' Draw lines across hole

  ' BorderStyle = dash
   pGraphBox.FillStyle = 1
   pGraphBox.DrawStyle = 0

   pGraphBox.Line (intOHrad, intCenter - intOHrad)-(intGW + intOHrad, intCenter -
intOHrad)
   pGraphBox.Line (0, intCenter)-(2 * intOHrad, intCenter)
   Point.X = 0: Point.Y = intCenter 'Set the start-point's coordinates
          LineTo pGraphBox.hdc, intOHrad + intGW, intCenter 'Draw a line from the
active point to the given point
   pGraphBox.Line (intOHrad, intCenter + intOHrad)-(intGW + intOHrad, intCenter +
intOHrad)
   End Sub

Private Sub FillGraphArray1()
   Dim intI As Integer 'index of graph
   Dim intZ As Integer ' range of grid for length
   Dim intR As Integer ' range of grid for radius
   Dim intStepz, intstepr As Integer
   Dim Ymodup As Integer
   Dim Ymoddw As Integer
   Dim Xmod As Integer
   intStepz = GetzGridStep
   intstepr = GetrGridStep
   For intI = 0 To intGW Step intStepz 'Cycle the values of the graph array
      For intR = 1 To ic
      Ymoddw = Fix(intCenter + (intR * intstepr * 1) + intSCrad)
      Ymodup = Fix(intCenter - (intR * intstepr * 1) - intSCrad)
      pGraphBox.ForeColor = vbBlue

       For intZ = intI - intStepz To intI  'Cycle the values of the graph array
      Xmod = Fix(intOHrad + intZ + intStepz)
      'Call SetGraphValue(rho(stepn, intR, 1, intZ))
      pGraphBox.PSet (Xmod, Ymodup)
      pGraphBox.PSet (Xmod, Ymoddw)
     'pGraphBox.Circle (Xmod, Ymodup), intstepz / 2, pGraphBox.ForeColor, , , 1
      'pGraphBox.Circle (Xmod, Ymoddw), intstepz / 2, pGraphBox.ForeColor, , , 1
      Next intZ

      Next intR


   Next intI
```

```
End Sub
Public Sub FillGraphArray2()
Dim intI As Integer 'index of graph
   Dim intZ As Integer ' range of grid for length
   Dim intR As Integer ' range of grid for radius
   Dim intet As Integer ' range of grid for radius
   Dim intStepz, intstepr As Integer
   Dim Ymodup As Integer
   Dim Ymoddw As Integer
   Dim Xmod As Integer
   intStepz = GetzGridStep
   intstepr = GetrGridStep
    For intI = 0 To intGW - intStepz Step intStepz 'Cycle the values of the graph array
      For intR = 1 To ic
      Ymoddw = Fix(intCenter + ((intR - 1) * intstepr * 1) + intSCrad)
      Ymodup = Fix(intCenter - (intR * intstepr * 1) - intSCrad)
     intet = 1 / 4 * jc
      If (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= 0 And phi(stepn, intR,
intet, Int(intI / intStepz)) <= (0.1)) Then
      pGraphBox.ForeColor = &H40&
      ElseIf (phi(stepn, intR, intet, Int((intI / intStepz)) >= (0.1) And phi(stepn, intR,
intet, Int(intI + intStepz) / intStepz)) <= (0.2)) Then
      pGraphBox.ForeColor = &H40C0&
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.2) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.3)) Then
      pGraphBox.ForeColor = &H8080&
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.3) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.4)) Then
      pGraphBox.ForeColor = &H404080
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.4) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.5) Then
      pGraphBox.ForeColor = vbBlue
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.5) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.6)) Then
      pGraphBox.ForeColor = vbRed
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.6) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.7)) Then
      pGraphBox.ForeColor = vbMagenta
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.7) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.8) Then
      pGraphBox.ForeColor = vbBlack
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.8) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.9) Then
      pGraphBox.ForeColor = vbGreen
      ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.9) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 1) Then
      pGraphBox.ForeColor = vbYellow
      End If

      For intZ = intI - intStepz To intI  'Cycle the values of the graph array
```

```
        Xmod = Fix(intOHrad + intZ + intStepz)
        'Call SetGraphValue(rho(stepn, intR, 1, intZ))
        'pGraphBox.PSet (Xmod, Ymodup)
        pGraphBox.PSet (Xmod, Ymoddw)
        'pGraphBox.Line (Xmod, Ymodup)-Step(intstepz / 2, intstepr / 2), , BF
        pGraphBox.Line (Xmod, Ymoddw)-Step(intStepz / 2, intstepr), , BF


       ' pGraphBox.Circle (Xmod, Ymodup), intstepz / 2, pGraphBox.ForeColor, , , 1
       ' pGraphBox.Circle (Xmod, Ymoddw), intstepz / 2, pGraphBox.ForeColor, , , 1
        Next intZ

      intet = 2 / 4 * jc
        If (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= 0 And phi(stepn, intR,
intet, Int(intI / intStepz)) <= (0.1)) Then
        pGraphBox.ForeColor = &H40&
        ElseIf (phi(stepn, intR, intet, Int((intI / intStepz)) >= (0.1) And phi(stepn, intR,
intet, Int(intI + intStepz) / intStepz)) <= (0.2)) Then
        pGraphBox.ForeColor = &H40C0&
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.2) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.3)) Then
        pGraphBox.ForeColor = &H8080&
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.3) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.4)) Then
        pGraphBox.ForeColor = &H404080
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.4) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.5) Then
        pGraphBox.ForeColor = vbBlue
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.5) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.6)) Then
        pGraphBox.ForeColor = vbRed
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.6) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.7)) Then
        pGraphBox.ForeColor = vbMagenta
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.7) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.8) Then
        pGraphBox.ForeColor = vbBlack
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.8) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.9) Then
        pGraphBox.ForeColor = vbGreen
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.9) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 1) Then
        pGraphBox.ForeColor = vbYellow
        End If

        For intZ = intI - intStepz To intI  'Cycle the values of the graph array
        Xmod = Fix(intOHrad + intZ + intStepz)
        pGraphBox.PSet (Xmod, Ymodup)
        pGraphBox.Line (Xmod, Ymodup)-Step(intStepz / 2, intstepr), , BF
        Next intZ
```

```
        Next intR
    Next intI
End Sub
Public Sub FillGraphArray3()
Dim intI As Integer 'index of graph
    Dim intZ As Integer ' range of grid for length
    Dim intR As Integer, intR2 As Integer ' range of grid for radius
    Dim intet As Integer ' range of grid for radius
    Dim intStepz, intstepr As Integer
    Dim Ymodup As Integer
    Dim Ymoddw As Integer
    Dim Xmod As Integer
    intStepz = GetzGridStep
    intstepr = GetrGridStep
    For intI = 0 To intGW - intStepz Step intStepz 'Cycle the values of the graph array
        For intR2 = -Int(intSCrad / (intOHrad - intSCrad) + 1) * ic To ic
        intR = IIf(intR2 > 0, intR2, ic)
        Ymoddw = Fix(intCenter + ((intR2 - 1) * intstepr * 1) + intSCrad)
        Ymodup = Fix(intCenter - (intR2 * intstepr * 1) - intSCrad)
      intet = 1 / 4 * jc
        If (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) > 0 And phi(stepn, intR,
intet, Int(intI / intStepz)) <= (0.1)) Then
        pGraphBox.ForeColor = &H40&
        ElseIf (phi(stepn, intR, intet, Int((intI / intStepz)) >= (0.1) And phi(stepn, intR,
intet, Int(intI + intStepz) / intStepz)) <= (0.2)) Then
        pGraphBox.ForeColor = &H40C0&
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.2) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.3)) Then
        pGraphBox.ForeColor = &H8080&
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.3) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.4)) Then
        pGraphBox.ForeColor = &H404080
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.4) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.5) Then
        pGraphBox.ForeColor = vbBlue
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.5) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.6)) Then
        pGraphBox.ForeColor = vbRed
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.6) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.7)) Then
        pGraphBox.ForeColor = vbMagenta
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.7) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.8) Then
        pGraphBox.ForeColor = vbBlack
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.8) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.9) Then
        pGraphBox.ForeColor = vbGreen
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.9) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 1) Then
        pGraphBox.ForeColor = vbYellow
```

```
        End If

        For intZ = intI - intStepz To intI  'Cycle the values of the graph array
        Xmod = Fix(intOHrad + intZ + intStepz)
        pGraphBox.PSet (Xmod, Ymoddw)
        pGraphBox.Line (Xmod, Ymoddw)-Step(intStepz / 2, intstepr), , BF
        Next intZ
      intet = 4 / 4 * jc
        If (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= 0 And phi(stepn, intR,
intet, Int(intI / intStepz)) < (0.1)) Then
        pGraphBox.ForeColor = &H40&
        ElseIf (phi(stepn, intR, intet, Int((intI / intStepz)) >= (0.1) And phi(stepn, intR,
intet, Int(intI + intStepz) / intStepz)) <= (0.2)) Then
        pGraphBox.ForeColor = &H40C0&
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.2) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.3)) Then
        pGraphBox.ForeColor = &H8080&
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.3) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.4)) Then
        pGraphBox.ForeColor = &H404080
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.4) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.5) Then
        pGraphBox.ForeColor = vbBlue
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.5) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.6)) Then
        pGraphBox.ForeColor = vbRed
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.6) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= (0.7)) Then
        pGraphBox.ForeColor = vbMagenta
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.7) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.8) Then
        pGraphBox.ForeColor = vbBlack
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.8) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 0.9) Then
        pGraphBox.ForeColor = vbGreen
        ElseIf (phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) >= (0.9) And
phi(stepn, intR, intet, Int((intI + intStepz) / intStepz)) <= 1) Then
        pGraphBox.ForeColor = vbYellow
        End If

        For intZ = intI - intStepz To intI  'Cycle the values of the graph array
        Xmod = Fix(intOHrad + intZ + intStepz)
        pGraphBox.PSet (Xmod, Ymodup)
        pGraphBox.Line (Xmod, Ymodup)-Step(intStepz / 2, intstepr), , BF
        Next intZ
        Next intR2
    Next intI
End Sub

Private Function GetrGridStep() As Integer
```

```
    GetrGridStep = mintStepr 'Return the X grid step amount
End Function

Private Function GettetaGridStep() As Integer
    GettetaGridStep = mintStepteta 'Return the X grid step amount
End Function

Public Sub SetrGridStep(ByVal intstepr As Integer)
    mintStepr = intstepr 'Set the r grid step amount
End Sub

Public Sub SettetaGridStep(ByVal intStepteta As Integer)
    mintStepteta = intStepteta 'Set the teta grid step amount
End Sub

Private Function GetzGridStep() As Integer
    GetzGridStep = mintStepz 'Return the z grid step amount
End Function

Private Function GethGridStep() As Integer
    GethGridStep = mintStepz 'Return the hcenter grid step amount
End Function

Public Sub SetzGridStep(ByVal intStepz As Integer)
    mintStepz = intStepz 'Set the z grid step amount
End Sub
Public Sub SethGridStep(ByVal intStepz As Integer)
    mintStepz = intStepz 'Set the h grid step amount
End Sub

Public Sub ModCleanUp()
    Set pGraphBox = Nothing 'clean up
End Sub
```

APPENDIX C: EXAMPLE OF PROGRAM OUTPUT FILE

Horizontal Well Simulation Run for: Adanga1OHP Run on : 5/11/2004 3:54:38 PM

Final Pack Efficiency = 74.60039 %

Toal Simulator Run Time = 8.889128 mins

The Settling Factor = No settling Effect

==

Time	Surface Injection Pressure	Pack Efficiency
(mins)	(Psia)	(%)
0.0	0.0	0.0
7.5	3252.7	0.0
15.0	3622.6	21.4
22.5	3620.5	48.6
30.0	3621.8	62.5
37.5	3622.2	65.8
45.0	3625.7	67.0
52.5	3621.2	70.9
60.0	3621.3	72.8
67.5	3621.4	73.9
75.0	3621.4	73.9
82.5	3621.4	74.1
90.0	3621.4	74.3
97.5	3621.4	74.3
105.0	3621.4	74.3
112.5	3621.4	74.3
120.0	3621.4	74.3
127.5	3621.4	74.3
135.0	3621.4	74.5
142.5	3621.4	74.5
150.0	3621.4	74.5
157.5	3621.4	74.5
165.0	3621.4	74.5
172.5	3621.4	74.5
180.0	3621.4	74.5
187.5	3621.4	74.5
195.0	3621.4	74.5
202.5	3621.4	74.5
210.0	3621.4	74.5
217.5	3621.4	74.5

225.0	3621.4	74.5
232.5	3621.4	74.5
240.0	3621.4	74.5
247.5	3621.4	74.5
255.0	3621.4	74.5
262.5	3621.4	74.5
270.0	3621.4	74.5
277.5	3621.4	74.6
285.0	3621.4	74.6
292.5	3621.4	74.6
300.0	3621.4	74.6
307.5	3621.4	74.6
315.0	3621.4	74.6
322.5	3621.4	74.6
330.0	3621.4	74.6
337.5	3621.4	74.6
345.0	3621.4	74.6
352.5	3621.4	74.6
360.0	3621.4	74.6
367.5	3621.4	74.6
375.0	3621.4	74.6
382.5	3621.4	74.6
390.0	3621.4	74.6
397.5	3621.4	74.6
405.0	3621.4	74.6
412.5	3621.4	74.6
420.0	3621.4	74.6
427.5	3621.4	74.6
435.0	3621.4	74.6
442.5	3621.4	74.6
450.0	3621.4	74.6

Time	Grid [i,j,k]	Average Pressure	Density	vz	vr	phi
(mins)	Grid [i,j,k](Psia)	(lbs/gal)	ft/sec	ft/sec	fraction	
0	[1,1,1]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,2]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,3]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,4]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,5]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,6]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,7]	4200.0	11.0	2.3	0.0	1.0

Time	Grid [i,j,k]	Average Pressure	Density	vz	vr	phi
(mins)	Grid [i,j,k](Psia)	(lbs/gal)	ft/sec	ft/sec	fraction	
0	[1,1,8]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,9]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,10]	4200.0	11.0	2.3	0.0	1.0
0	[1,1,11]	4200.0	11.0	2.3	0.0	1.0
0	[4,8,5]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,6]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,7]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,8]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,9]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,10]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,11]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,12]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,13]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,14]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,15]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,16]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,17]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,18]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,19]	4200.0	11.0	0.0	0.0	1.0
0	[4,8,20]	4200.0	11.0	0.0	0.0	1.0
7.5	[1,1,1]	4208.0	2.3	2.3	0.0	1.0
7.5	[1,1,2]	4164.5	1.1	2.4	0.0	1.0
7.5	[1,1,3]	4121.0	1.1	2.4	0.0	1.0
7.5	[1,1,4]	4077.6	1.1	2.4	0.0	1.0
7.5	[1,1,5]	4034.1	1.1	2.4	0.0	1.0
7.5	[1,1,6]	3990.7	1.1	2.4	0.0	1.0
7.5	[1,1,7]	3947.2	1.1	2.5	0.0	1.0
7.5	[1,1,8]	3903.7	1.1	2.5	0.0	1.0
7.5	[1,1,9]	3860.3	1.1	2.5	0.0	1.0
7.5	[1,1,10]	3816.8	1.1	2.5	0.0	1.0
7.5	[1,1,11]	3773.3	2.3	2.6	0.0	1.0
22.5	[2,1,16]	3564.0	11.4	2.7	0.0	0.0
22.5	[2,1,17]	3520.6	11.3	2.7	0.0	0.0
22.5	[2,1,18]	3477.1	11.2	2.7	0.0	0.0
22.5	[2,1,19]	3433.6	11.1	2.7	0.0	0.0
22.5	[2,1,20]	3390.2	3.1	2.8	0.0	0.6
22.5	[2,2,1]	4200.0	20.7	2.3	0.0	0.0
22.5	[2,2,2]	4129.1	8.9	2.4	0.0	1.0
22.5	[2,2,3]	4085.6	8.9	2.4	0.0	1.0

Time	Grid [i,j,k]	Average Pressure	Density	vz	vr	phi
(mins)	Grid [i,j,k](Psia)	(lbs/gal)	ft/sec	ft/sec	fraction	
22.5	[2,2,4]	4042.1	8.9	2.4	0.0	1.0
22.5	[2,2,5]	3998.7	8.9	2.4	0.0	1.0
22.5	[2,2,6]	3955.2	8.9	2.4	0.0	1.0
22.5	[2,2,7]	3911.7	8.1	2.5	0.0	1.0
22.5	[2,2,8]	3868.3	9.1	2.5	0.0	1.0
22.5	[2,2,9]	3824.8	8.9	2.5	0.0	1.0
22.5	[2,2,10]	3781.3	9.1	2.5	0.0	1.0
22.5	[2,2,11]	3737.9	6.2	2.6	0.0	1.0
22.5	[2,2,12]	3694.4	1.7	2.6	0.0	0.8
22.5	[2,2,13]	3651.0	5.1	2.6	0.0	0.4
22.5	[2,2,14]	3607.5	10.1	2.6	0.0	0.0
22.5	[2,2,15]	3564.0	11.4	2.7	0.0	0.0
22.5	[2,2,16]	3520.6	11.3	2.7	0.0	0.0
22.5	[2,2,17]	3477.1	20.7	2.7	0.0	0.0
22.5	[2,2,18]	3433.6	8.5	2.7	0.0	1.0
22.5	[2,2,19]	3390.2	11.2	2.7	0.0	0.0
22.5	[2,2,20]	3346.7	8.3	2.8	0.0	1.0
22.5	[2,3,1]	4217.6	20.7	2.3	0.0	0.0
22.5	[2,3,2]	4085.6	8.9	2.4	0.0	1.0
22.5	[2,3,3]	4042.1	8.9	2.4	0.0	1.0
22.5	[2,3,4]	3998.7	8.9	2.4	0.0	1.0
22.5	[2,3,5]	3955.2	8.9	2.4	0.0	1.0
22.5	[2,3,6]	3911.7	8.9	2.4	0.0	1.0
22.5	[2,3,7]	3868.3	8.9	2.5	0.0	1.0
22.5	[2,3,8]	3824.8	8.9	2.5	0.0	1.0
22.5	[2,3,9]	3781.3	8.9	2.5	0.0	1.0
22.5	[2,3,10]	3737.9	5.5	2.5	0.0	1.0
22.5	[2,3,11]	3694.4	1.9	2.6	0.0	0.8
22.5	[2,3,12]	3651.0	5.2	2.6	0.0	0.4
22.5	[2,3,13]	3607.5	10.1	2.6	0.0	0.0
22.5	[2,3,14]	3564.0	11.3	2.6	0.0	0.0
22.5	[2,3,15]	3520.6	11.2	2.7	0.0	0.0
22.5	[2,3,16]	3477.1	11.2	2.7	0.0	0.0
22.5	[2,3,17]	3433.6	11.1	2.7	0.0	0.0
22.5	[2,3,18]	3390.2	11.0	2.7	0.0	0.0
22.5	[2,3,19]	3346.7	10.9	2.7	0.0	0.0
22.5	[2,3,20]	3303.2	1.4	2.8	0.0	0.7
22.5	[2,4,1]	4259.4	20.7	2.3	0.0	0.0
22.5	[2,4,2]	4042.1	8.9	2.4	0.0	1.0

Time	Grid [i,j,k]	Average Pressure	Density	vz	vr	phi
(mins)	Grid [i,j,k](Psia)	(lbs/gal)	ft/sec	ft/sec	fraction	
22.5	[2,4,3]	3998.7	8.9	2.4	0.0	1.0
22.5	[2,4,4]	3955.2	8.9	2.4	0.0	1.0
22.5	[2,4,5]	3911.7	8.9	2.4	0.0	1.0
22.5	[2,4,6]	3868.3	8.9	2.4	0.0	1.0
22.5	[2,4,7]	3824.8	8.1	2.5	0.0	1.0
22.5	[2,4,8]	3781.3	9.1	2.5	0.0	1.0
22.5	[2,4,9]	3737.9	8.9	2.5	0.0	1.0
22.5	[2,4,10]	3694.4	9.1	2.5	0.0	1.0
22.5	[2,4,11]	3651.0	2.7	2.6	0.0	0.4
22.5	[2,4,12]	3607.5	18.8	2.6	0.0	0.0
22.5	[2,4,13]	3564.0	8.8	2.6	0.0	1.0
22.5	[2,4,14]	3520.6	2.3	2.6	0.0	0.4
22.5	[2,4,15]	3477.1	11.1	2.7	0.0	0.0
22.5	[2,4,16]	3433.6	11.0	2.7	0.0	0.0
22.5	[2,4,17]	3390.2	20.5	2.7	0.0	0.0
22.5	[2,4,18]	3346.7	8.3	2.7	0.0	1.0
22.5	[2,4,19]	3303.2	11.0	2.7	0.0	0.0
22.5	[2,4,20]	3259.8	8.0	2.8	0.0	1.0
22.5	[2,5,1]	4216.0	20.7	2.3	0.0	0.0
22.5	[2,5,2]	3998.7	8.9	2.4	0.0	1.0
22.5	[2,5,3]	3955.2	8.9	2.4	0.0	1.0
22.5	[2,5,4]	3911.7	8.9	2.4	0.0	1.0
22.5	[2,5,5]	3868.3	8.9	2.4	0.0	1.0
22.5	[2,5,6]	3824.8	8.9	2.4	0.0	1.0
22.5	[2,5,7]	3781.3	8.9	2.5	0.0	1.0
22.5	[2,5,8]	3737.9	5.7	2.5	0.0	1.0
22.5	[2,5,9]	3694.4	2.1	2.5	0.0	0.8
22.5	[2,5,10]	3651.0	5.3	2.5	0.0	0.4
22.5	[2,5,11]	3607.5	10.1	2.6	0.0	0.0
22.5	[2,5,12]	3564.0	11.3	2.6	0.0	0.0
22.5	[2,5,13]	3520.6	11.2	2.6	0.0	0.0
22.5	[2,5,14]	3477.1	11.1	2.6	0.0	0.0
22.5	[2,5,15]	3433.6	11.0	2.7	0.0	0.0
22.5	[2,5,16]	3390.2	10.9	2.7	0.0	0.0
22.5	[2,5,17]	3346.7	10.8	2.7	0.0	0.0
22.5	[2,5,18]	3303.2	10.8	2.7	0.0	0.0
187.5	[3,6,6]	3781.3	45.8	2.4	0.0	0.0
187.5	[3,6,7]	3737.9	3.9	2.5	0.0	0.8
187.5	[3,6,8]	3694.4	5.5	2.5	0.0	0.4

Time	Grid [i,j,k]	Average Pressure	Density	vz	vr	phi
(mins)	Grid [i,j,k](Psia)	(lbs/gal)	ft/sec	ft/sec	fraction	
187.5	[3,6,9]	3651.0	22.8	2.5	0.0	0.0
187.5	[3,6,10]	3607.5	100.7	2.5	0.0	0.0
187.5	[3,6,11]	3564.0	12.1	2.6	0.0	0.0
187.5	[3,6,12]	3520.6	8.0	2.6	0.0	0.0
187.5	[3,6,13]	3477.1	13.7	2.6	0.0	0.0
187.5	[3,6,14]	3433.6	13.8	2.6	0.0	0.0
187.5	[3,6,15]	3390.2	52.4	2.7	0.0	0.0
187.5	[3,6,16]	3346.7	13.5	2.7	0.0	0.0
187.5	[3,6,17]	3303.2	71.2	2.7	0.0	0.0
187.5	[3,6,18]	3259.8	29.7	2.7	0.0	0.0
187.5	[3,6,19]	3216.3	55.7	2.7	0.0	0.0
187.5	[3,6,20]	3172.9	71.9	2.8	0.0	0.0
187.5	[3,7,1]	4222.9	33.1	2.3	0.0	0.0
187.5	[3,7,2]	3911.7	79.3	2.4	0.0	0.0
187.5	[3,7,3]	6084.3	8.8	2.4	0.0	1.0
187.5	[3,7,4]	3824.8	27.2	2.4	0.0	0.0
187.5	[4,6,20]	3172.9	70.3	2.8	0.0	0.0
187.5	[4,7,1]	4292.6	33.1	2.3	0.0	0.0
187.5	[4,7,2]	3911.7	79.3	2.4	0.0	0.0
187.5	[4,7,3]	3868.3	26.7	2.4	0.0	0.0
187.5	[4,7,4]	3824.8	43.6	2.4	0.0	0.0
187.5	[4,7,5]	3781.3	3.9	2.4	0.0	0.8
187.5	[4,7,6]	3737.9	5.7	2.4	0.0	0.4
187.5	[4,7,7]	3694.4	20.3	2.5	0.0	0.0
187.5	[4,7,8]	3651.0	75.9	2.5	0.0	0.0
187.5	[4,7,9]	3607.5	12.1	2.5	0.0	0.0
187.5	[4,7,10]	3564.0	8.1	2.5	0.0	0.0
187.5	[4,7,11]	3520.6	13.6	2.6	0.0	0.0
187.5	[4,7,12]	3477.1	13.3	2.6	0.0	0.0
187.5	[4,7,13]	3433.6	44.4	2.6	0.0	0.0
187.5	[4,7,14]	3390.2	54.0	2.6	0.0	0.0
187.5	[4,7,15]	3346.7	16.3	2.7	0.0	0.0
187.5	[4,7,16]	3303.2	13.2	2.7	0.0	0.0
187.5	[4,7,17]	3259.8	81.9	2.7	0.0	0.0
187.5	[4,7,18]	3216.3	33.8	2.7	0.0	0.0
187.5	[4,7,19]	3172.9	66.2	2.7	0.0	0.0