AN ARTIFICIAL INTELLIGENCE APPROACH
TO WELL TEST INTERPRETATION

A Report Submitted to the Department of Petroleum Engineering of Stanford University in Partial Fulfillment of the Requirements for the Degree of Master of Science

by
Olivier Allain
June 1987
Acknowledgements

I would like to express my gratitude to Dr. Roland Horne for his suggestions and guidance during this study.

Support provided by Flopetrol-Johnston (now Schlumberger Perforating and Testing Services) in the form of a scholarship, is also gratefully acknowledged.
Abstract

The improvements in well test analysis from the former "straight line approach" to its current state are strongly related to the use of computers in this field. Automation of the interpretation procedure has been achieved by automating the type-curve matching. This automation not only speeds up the analysis but also increases its reliability by providing a match free of any subjective consideration of goodness. Subjectivity is however still present in the choice of the interpretation model used to match the data. The objective of this study is to automate this choice.

A log-log plot of the pressure derivative constitutes an appropriate diagnostic tool to choose an analytical model for given data. It yields distinct features for different flow regimes, which an expert can recognize and relate to specific models. The complete interpretation model is then obtained by combining these various components. Using the log-log plot enables the expert to visually replace the original data by a symbolic representation, describing the response like a succession of shapes. If the data are smoothed, we can easily reproduce this substitution.

We will assume that such a smoothing can be performed and therefore concentrate on choosing an interpretation model for smoothed data only.

Artificial intelligence language and techniques are used to achieve an adequate representation of the response as well as to simulate the reasoning processes involved in the identification of a model. The knowledge of the producible theoretical responses is included using tree structures, and can therefore be easily augmented with new models.

Some correlations are also presented which relate the size of the shapes produced by particular models to the value of the parameters associated with them. These relations are used to give a first match from which an automated type-curve matching can start.
Table of Contents

- ACKNOWLEDGEMENTS ......................................................... i.
- ABSTRACT ........................................................................ ii.
- TABLE OF CONTENTS .......................................................... iii.
- LIST OF FIGURES ................................................................. iv.
- INTRODUCTION ..................................................................... 1.
  - 1. ADOPTED APPROACH ......................................................... 4.
  - 2. SHAPE EXTRACTION ........................................................... 6.
  - 3. CORRELATION OF RESPONSE SHAPE TO RESERVOIR CHARACTERISTICS ......................................................... 10.
  - 4. QUANTITATIVE DESCRIPTION ............................................... 16.
  - 5. APPLICATIONS ................................................................. 18.
  - 6. CONCLUSION ................................................................. 23.
- FIGURES ................................................................................. 24.
- BIBLIOGRAPHY ....................................................................... 42.
- APPENDICES
  - APPENDIX A Example 1............................................................. 44.
  - APPENDIX B: Example 2 .......................................................... 45.
  - APPENDIX C: program ............................................................. 47.
List of Figures

1. $C_{deS} = 100$ infinite homogeneous reservoir .......................... 24.
2. Initial tree for the interpretation ..................................... 25.
4. MODEL3 ........................................................... 27.
5. MODEL2 and MODEL1 ............................................. 28.
6. Tree for boundary effects ............................................ 29.
7. Example of boundary effects ........................................ 30.
8. Correlation for $C_{deS}$ ............................................... 31.
9. Correlation for time match ........................................... 32.
10. Correlation for $\omega$ .............................................. 33.
11. Example 1 .......................................................... 34.
12. Example 1: path followed ........................................... 35.
13. Example 1: match obtained ........................................... 36.
14. Example 2 .......................................................... 37.
15. Example 2: path followed ........................................... 38.
17. Example 1 and Example 2: comparison (a) .................... 40.
18. Example 1 and Example 2: comparison (b) .................... 41.
Automated model identification is a logical continuation in the evolution of well test interpretation techniques. This evolution can be decomposed into four main steps: straight line analysis, inverse problem approach, introduction of the derivative and automation of the type-curve matching.

Transient test interpretation techniques have formerly been reduced to the identification of characteristic regimes which produce a straight line on the appropriate graph: $p$ versus $\log t$ for radial flow, $p$ versus $\sqrt{t}$ for linear flow, etc. This method is particularly convenient for hand analysis (and was justified when the current computing facilities were not available). It fails to make use of all the data and can result in significant errors.

With the developments of analytical models, the interpretation procedure may be viewed as the resolution of an inverse problem for the reservoir and well system ($S$). Inasmuch as $S$ is recognizable by its response $O$ (variation of pressure vs elapsed time) to a known input $I$ (flowrate history), the solution is found as an analytical model $S'$ whose response $O'$ to $I$ is the closest possible to $O$. The choice of $S'$ is directed by the shape of the data on a log-log plot (sometimes called a "diagnostic plot"). Unlike the "straight line approach", this procedure allows us to treat the data as a whole, not only looking for some particular regimes. Moreover, it adds confidence in the determination of such particular regimes (like infinite acting radial flow for instance). However, specialized plots are still necessary.

The next step in the evolution of the interpretation was the introduction of the pressure derivative\cite{14,17}. First, sections which produce a straight line on specialized plots are of constant derivative slope, they can therefore be all diagnosed on the same plot. Second, the derivative allows the determination of some other characteristic shapes, like double porosity transitions, with greater definition.

With the pressure derivative, it is easier to make the most appropriate choice of an analytical model. For constant rate tests, the analysis also becomes a single plot procedure\cite{20}:

- Pressure and pressure derivative are plotted on a log-log scale. This plot (diagnostic plot) enables us to distinguish various flow regimes in the behavior.

- The succession of flow regimes identified determines the choice of a theoretical model, i.e. a qualitative representation of the system.
Matching the response of the theoretical model with the data confirms the diagnosis and provides a quantitative representation of the system.

An improvement in this procedure has been gained by automating the type-curve matching. Linear regression algorithms were first used, and more recently, non-linear ones. This automation significantly speeds up the analysis, therefore allowing a real-time interpretation during the monitoring of the well test. By providing a match free of any subjective consideration of goodness, it improves the reliability. However, the reliability of the analysis still depends on a subjective interpretation: a judgement must still be made as to the appropriate reservoir model, based on the identification of the shapes present on the pressure derivative.

Whereas type-curve matching is only a numerical procedure, the choice of the analytical model does not involve computations. A log-log plot of the pressure derivative vs. elapsed time yields a limited number of characteristic features for the various components of the interpretation model that are easy to recognize. This recognition is achieved by simply looking at the derivative plot. The complete interpretation model is then obtained by combining the various components. The diagnosis made on the log-log plot seems at first much more appropriate to human beings than to computers (whereas type-curve matching is not). It requires reasoning capabilities as well as the ability to perceive a curve not as a set of digitized data but as a succession of shapes.

The use of computers to exhibit human behavior and simulate reasoning processes has been greatly enhanced with the evolution of Artificial Intelligence (usually referred to as AI). A very general definition of AI is the following:

"Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better." This definition is certainly too vague to stand for a promise "a priori" of success in the application of AI to well-test interpretation. However, some well-known examples show that AI has been successful in a lot of domains: mass spectrum interpretation, computer configuration, diagnosis of infectious blood diseases, mineral deposit prospecting, discovery in mathematics. Other systems have been developed in the oil industry, for log interpretation and dip-meter interpretation. The common feature of these programs is that they all include reasoning processes and that these processes are based on symbolic manipulations. Computer languages such as LISP are suited for programming computers to manipulate symbols.

It seems appropriate to apply AI techniques to the choice of an interpretation model for given data for the following reasons. First, using the pressure derivative plot is a way of perceiving the data in a symbolic fashion. With the use of symbol manipulation languages such as LISP, the curve can be represented appropriately.
Second, most of the information required to choose the model is qualitative information. Very little quantitative information is required. AI programming languages and techniques provide proper ways of representing qualitative information.

The artificial intelligence approach should permit well test interpretation to be fully automated and therefore lead to the creation of an expert system of interpretation. Such a system would be of great importance, not only commercially. It would give well test analysis a reliability at present founded on a subjective diagnosis made on a log-log plot. Moreover, the solution to an interpretation is generally not unique. An automatic analysis might come up with solutions not even considered by the human expert.
1. ADOPTED APPROACH

The interpretation procedure, described in the introduction as the resolution of an inverse problem, is based on the assumption that the analytical models available are sufficient to reproduce any field case. Or differently, any field test which cannot be reproduced using the available analytical models cannot be interpreted.

Accepting this basic assumption, there are two necessary conditions which must be met for the interpretation to be possible. An analytical model is chosen by recognizing a shape on the derivative plot that is associated with an effect characteristic of that model. For instance, a unit slope at early time is indicative of wellbore storage. In order to recognize something, one must already know it: the main requirement for performing the interpretation is to know all (or at least a lot of) the possible analytical responses.

This knowledge is available to the expert as a symbolic representation. The expert does not know a model by the analytical expression associated with it nor as a list of digitized data. Rather, he/she will mentally substitute a specific response by a symbolic representation which characterizes it. This representation could be seen as the description he/she would give to define the pressure derivative on a log-log plot. Wellbore storage and skin for a damaged well in an infinite homogeneous medium can be described by "A hump starting with a unit slope at early time and stabilization after the end of the hump".

With such a representation for each model, the interpretation of a given test proceeds as follows. First, the data are replaced by a similar representation, formed using the same vocabulary (or the same symbols). Second, this representation is compared to the theoretical ones and eventually matched.

Because a typical response is the combination of three different regimes, early time, average time, late time, the expert will try to differentiate these sections first and then interpret them independently. Therefore, instead of being one set of representations for theoretical models, we can imagine that the expert's knowledge is structured in three different lists.

In order to simulate the same expert behavior, we first need to make available to the program as many response representations as possible - possibly all the theoretical ones. Then, to interpret a given test, the program should transform it to a similar representation. This substitution is done instantaneously by the expert through the process of vision. For a smoothed curve it is straightforward. For a noisy one, it requires some expertise to know that part of the shape is noise, some other part is not. However, it can still be done instantaneously. Since humans have
This ability to perceive digitized data like an analog signal even when they are noisy. giving a symbolic representation to the data is a natural and minor task for them. This is not true for computers.

Two possible approaches can be suggested. The first one is to try and extract the shapes from the noisy data, using a pattern recognition technique. This approach is certainly the most appropriate to simulate vision, but also the most computationally intensive. The other one is to start by smoothing the data and then perform the shape extraction directly and simply. This second approach was chosen.

Although smoothing is necessary, some flexibility has also been introduced in the program making the shape extraction - as will be explained in section 2.

Once a model has been found by matching the symbolic representation of the data to a theoretical representation, we need to match the data themselves. Automated type-curve matching can be used for this task provided that along with the model, a first estimate of the parameter values be given. To give this estimate, we must develop relations between the parameters of a model and the amplitude of the shapes it exhibits. Such relations have been developed for some of the models and added to the program.
2. SHAPE EXTRACTION

It is assumed for this task that the data have been properly smoothed, that is all shapes present correspond to physical phenomena in the reservoir and all shapes due to noise have been removed. This requirement does actually not need to be true for small amplitude shapes.

The goal is to replace the response (considered in terms of pressure derivative) by a symbolic representation which is a list of "known" shapes. "Known" meaning that they can be produced by one of the available models and therefore are present in the list of theoretical representations.

The first step is therefore to define which shapes we will be looking for, the leading requirement being that they allow us to represent any theoretical response. These shapes are limited in number and can be divided into four main categories:

1. Straight sections

They correspond to regimes such as radial flow, linear flow, pseudo steady state flow.

2. Extrema

Extrema can be due to wellbore storage and skin for a maximum, to heterogeneous behavior for a minimum.

3. Transitions

Transitions are those portions which are followed during a change from one regime to another. They often exhibit an inflexion.

4. Upward or Downward trends

Upward or downward trends can appear at the end of the derivative due to boundary effects.

Defining these shapes can be viewed as defining the necessary vocabulary to describe with words all theoretical responses. The description must be such that it distinguishes what needs to be distinguished. For instance, a straight portion does not have the same meaning if the slope is 1 or if it is 1/2. Therefore, the representation of straight sections must include the slope.
PROCEDURE

The input to the program is a set of data $\log(\text{pressure derivative})$ vs. $\log(\Delta t)$. The initial data are first changed to a list of segments, two consecutive points defining a segment. The segments have the following structure:

$$\{n \; \text{slope} \; (x_1 \cdot y_1) \; (x_2 \cdot y_2)\}$$

- $n$: number of points bound (2 initially).
- slope: slope of the segment.
- $(x_1 \cdot y_1)$: starting point.
- $(x_2 \cdot y_2)$: end point.

Segments are then extended according to a tolerance parameter ($\epsilon$). If two consecutive segments have slopes which do not differ by more than $\epsilon$, they are merged into a single segment whose slope is chosen as the average of the previous slopes. Each time a segment is merged, $n$ is incremented by one.

At the end of this step, we have achieved a continuous description of the data on which we can perform the shape extraction. The purpose of this extraction is to replace the response by a similar representation to the one we mentally form by looking at the pressure derivative curve. We parse the response looking for analytical features which will enable us to recognize the four main types of shapes listed earlier. When a given feature is found, it is replaced by a specific representation. The response is replaced by the combination of these various representations.

The analytical features sought in the list of segments are:

- **MAXIMUM**
  - It indicates the presence of a hump on the response, which for instance can be due to wellbore storage and skin.
  - representation: $(\text{MAXIMUM}(z \cdot y))$
  - $(z \cdot y)$: location

- **MINIMUM**
  - It indicates the presence of a trough, like a double porosity transition for instance.
  - representation: $(\text{MINIMUM}(z \cdot y))$
  - $(z \cdot y)$: location

- **INFLEXION**
Inflexions will allow us to recognize transitions due to boundary effects, to heterogeneous behavior, etc. A distinction is made between upward, downward and flat inflexions.

representation: \( (\text{INFLEXION } (z \cdot y) \text{ trend}) \)

\((z \cdot y)\): location

trend: UP, DOWN or FLAT.

A segment is associated with an inflexion if:

\[
(slope - ante.slope)x(post.slope - slope) < 0
\]

Where \(slope\) is the slope of the segment \(ante.slope\) the slope of the previous segment, \(post.slope\) the slope of the next segment:

- STRAIGHT SECTION

representation: \( (\text{STRAIGHT slope } (x_1 \cdot y_1) (x_2 \cdot y_2)) \)

\((x_1 \cdot y_1)\): starting point.

\((x_2 \cdot y_2)\): end point.

A segment is said to be a straight section if the number of points it binds \((n)\) is greater than \(*\text{number - points}*\) (which can be adjusted). If a segment diagnosed as "STRAIGHT has a slope less than \(*\epsilon - slope*\) (in absolute value), it is differentiated:

representation: \( (\text{STEP slope } (x_1 \cdot y_1) (x_2 \cdot y_2)) \)

After the list of segments has been parsed, if the last shape found does not correspond to the last segment, a final trend is defined and added to the list of shapes. This is done only if the trend is upward on downward.

representation: \( (\text{UTREND slope}) \) or \( (\text{DTREND slope}) \)

Figure 1 is the pressure derivative for a pressure response generated using the model representing flow with wellbore storage and skin in an infinite homogeneous medium with \(CDe^{28} = 100\) (generated using 9 points per log cycle). The following representation was obtained:

\[
((\text{MAXIMUM } (0.699 \cdot 0.095)) \ (\text{INFLEXION } (1.238 \cdot 0.061) \ DOWN) \\
(\text{STEP } - 3.436E - 4 \ (2.477 \cdot 0.286) \ (4.778 \cdot -0.301)))
\]

with \(*\epsilon - slope* = 0.01

\(*\epsilon - slope* = 0.01
With the functions described above, the program can substitute a symbolic representation which may then be used for model identification in place of the original data. In section 1 we defined this representation as a qualitative description of the pressure derivative curve on a log-log plot.

The representation chosen includes more than simply a qualitative description of the response. However, information like the location of an inflexion or the length of a straight section are not used in the model identification. They will be used after the identification has been performed to estimate the value of the parameters associated with the model. This estimation will be described in section 4.

**FLEXIBILITY**

Several adjustable parameters provide the shape extraction with flexibility:
*epsilon*: used for the segment extension. If it is increased, the program will not diagnose small humps or troughs which could remain after smoothing.
*number - points*: which determines whether a portion can be considered as straight or not.
3. CORRELATION OF RESPONSE SHAPE TO RESERVOIR CHARACTERISTICS

Using the procedure described in section 2, we now have a way to substitute a symbolic representation but the program does not yet have the knowledge of the theoretical responses.

A first approach could be to build a list of all producible theoretical curves. A test could then be analyzed by simply selecting representations systematically until one is found which matches the data representation. This procedure would be a top to bottom procedure, generating theoretical responses until the solution is found.

This is certainly not the way an expert will perform the interpretation. His/her procedure is rather from bottom to top: from the presence of given features on the response, he/she will proceed backward to the model. More precisely, it is likely that an expert will actually perform three different interpretations, one for early time, one for average time, one for late time. In fact, if we could find a way to differentiate at first these three regimes in the representation, we could then interpret them independently. The problem is how to differentiate them without interpreting them.

For instance, let us take the simple case of wellbore storage and skin for a damaged well in an infinite reservoir. To interpret the early time part of the data, we must recognize the time at which the flow regime changes to something different. There are two ways to do this:

1. Recognize early time as being wellbore storage and skin: then we know when it stops.

2. Recognize radial flow as a stabilization on the derivative: then we know that early time ends when radial flow starts.

From this example, we see that differentiation and interpretation are mixed. This is due to the fact that the number of basic shapes appearing on a pressure derivative is very limited compared to the number of producible responses. From this discrepancy, it is clear that a given shape has a lot of possible meanings and that the distinction must be syntactic, that is, it is related to the presence (or absence) of other shapes. Since this statement is true for all shapes appearing on the curve, it seems at first that the interpretation is only possible by having a global vision of the response at once.

Although the pressure derivative plot provides the expert with this global vision, it is not possible to achieve it with a computer. Therefore we must find another way
of starting the interpretation. We will still try to differentiate within the response the main three regimes but the procedure will be different from that of the expert.

Let us consider again the hypothetical list of all producible shapes and imagine that we organize it by similarities and differences. For instance, we could notice that a no flow boundary will create an inflexion (between \(0.5\) and \(1\)). Conversely, if no inflexion is found on the data representation we can infer that the effect of a no flow boundary has not occurred during the response. Based on such properties for categories of curves, we can decide on tests by which we will interpret the response. The effect of each test is to reduce the set of admissible models (or the search space). If we were to use this procedure to the end, the solutions would be obtained when all tests have been applied.

This set of tests can be organized in one or more "tree" structures.

**FIRST TREE**

The procedure described above is applied to the response until enough information is obtained to treat regimes separately. The tests are organized in a tree structure shown on Figure 2.

- **Node NO**: The first test is to check whether or not an inflexion is present on the curve; and was chosen for the following reason: wellbore storage and skin (for a damaged well), transitions due to heterogeneous behavior, and most of the effects due to no flow boundaries produce inflexions. If no inflexion is found then we can significantly reduce the search space.

- **Node N2**: No inflexion is found.
  
  If the early time part exhibits a hump due to wellbore storage, we know that this hump cannot be developed. Otherwise it would show an inflexion. The presence of this hump implies that there is a maximum.

- **Node N4**: There is a maximum.
  
  This is the case discussed above, it will be refered to as MODEL4 and is shown on Figure 3.

- **Node N5**: There is no maximum.
  
  The early time section does not exhibit a hump. Except if the test is very short, it is likely that early time corresponds to a low \(C_{De28}\) or a fracture. Then, any effect which does not produce an inflexion is admissible. The different possibilities are grouped under MODEL5 and shown on Figure 3.

Now we go back to Node NO assuming this time that an inflexion has been found.
o **Node N1**: An inflexion is found.

It can accompany any of the following effects: end of hump due to wellbore storage and skin, double porosity transition or boundary effects. If no maximum is present on the curve, then the early time cannot exhibit a hump.

o **Node N8**: No maximum found on the curve.

The possibilities associated with this case are grouped under MODEL3 and shown on Figure 4.

o **Node N3**: A maximum is found.

If the reservoir exhibits a double porosity behavior, the transition might start before the end of wellbore storage, and the hump will be followed not by radial flow but by the bottom of the transition. This case is such that a minimum will immediately follow the hump. The same shape will be obtained if a boundary effect occurs right after the hump.

o **Node N6**: A minimum follows the hump.

This is the case discussed above. It is grouped under MODEL1 and shown on Figure 5.

o **Node N7**: No minimum following the hump.

In this case, the regime following the hump is necessarily radial flow. The different possibilities are referred to as MODEL2 and shown on Figure 5.

This concludes the first tree. Once a terminal node is reached, the corresponding MODEL is called and through it different treatments applied to different sections of the representation.

**DIFFERENT REGIMES**

As was outlined before, the purpose of the first tree is to facilitate the differentiation of several regimes on the curve. For some of the models no further treatment is required, and the response is fully decomposed (MODEL4): for the majority of them we simply know when the early time section ends. To treat the different regimes, we will use the following procedure:

**Early Time**

Thanks to the initial parsing, we can remove from the representation the shapes corresponding to this regime. If a hump was diagnosed, then we know that it is due to wellbore storage and skin: no further analysis is required for the moment. If there is no maximum, we must decide between a low \( C_D e^{2s} \) or a fracture, or try
both possibilities. If the beginning of the response exhibits a straight line (defined symbolically as "STRAIGHT"). and its slope is higher than $1/2$, wellbore storage and skin is chosen. If the slope is close to $1/2$ both might be tried.

   If there was no straight line at the beginning, the same treatment is applied to the longest segment present.

Heterogeneous Behavior

   In the case of transient double porosity, the transition can be mistaken for the effect of a sealing fault. The difference of levels between the two stabilizations will however be different and can be used as a test for any boundary effect diagnosed.

   For pseudosteady state double porosity, the transition has a specific shape different from a boundary effect and characterized by the presence of a minimum. Moreover, when it occurs, this transition is superimposed on the homogeneous response (this is true on the derivative, not on the pressure). After going through the first tree we were able to remove the early time section: if we now remove a possible double porosity transition, the remaining list can only contain radial flow and possibly boundary effects.

Boundary Effects

   A specialized tree is used for interpreting boundary effects, which is shown on Figure 6. Some of the leaves can be followed more than once, which corresponds to the fact that several faults can influence the response successively with similar effects.

   It is assumed when we start in the tree that any shape which is not due to boundaries has been removed. In particular, radial flow must have been removed.

- Node B1

   If no shapes remain in the list, then no boundary effects are present and the interpretation is finished as far as the choice of the model is concerned. Otherwise, the three following shapes can occur: an upward trend, an upward inflexion, a downward trend.

- Node B2: An upward trend is found.

   This trend can be associated to linear flow or pseudo steady state flow according to its slope. If the trend is due to the effect of a sealing fault, then the test must have been stopped before the curve tends to a new stabilization. Otherwise there would be an upward inflexion instead of an upward trend. B2 is a terminal node.

- Node B4: A downward trend is found.
This trend is due to a constant pressure boundary (drawdown). It is possible that no flow boundaries be also present and will influence the response later. The fact that the derivative starts by dropping ensures that the constant pressure fault is the closest one. No further treatment is done in this case. \( B4 \) is a terminal node.

- **Node B3**: An upward inflexion is found.
  This inflexion accompanies the effect of one (or several) no flow boundaries. It can be followed by a stabilization of the derivative, a maximum if the curve drops afterwards due to a constant pressure fault. or a flat inflexion if another sealing fault is seen.

- **Node B6**: A maximum is found
  This implies that the derivative is dropping due to a constant pressure fault. No further analysis is performed. \( B6 \) is a terminal node.

- **Node B5**: A step or a flat inflexion is found.
  It corresponds to the effect of one (or several) sealing fault(s). If it is a step, then the curve can either go up or down: Nodes \( B2, B3 \) or \( B4 \). If it is an inflexion the curve can only go up: Nodes \( B3 \) or \( B2 \).

Some typical boundary effects are shown on Figure 7 along with the 'corresponding path followed in the tree.

**TREATMENT ASSOCIATED WITH EACH MODEL**

This part describes the effects of the functions associated with the terminal nodes in the initial tree.

- **MODEL1** (Figure 5)
  Early time is taken to be until the first inflexion on the response. The minimum is either considered as radial flow or the bottom of a double porosity transition. Then boundary effects are sought.

- **MODEL2** (Figure 5)
  Early time is taken to be until the first inflexion on the response. The shape following the minimum (step or inflexion) is taken as radial flow. A double porosity is sought within the remaining part. After it (or after radial flow if no transition is found) boundary effects are sought.

- **MODEL3** (Figure 4)
  If there is a step present before the inflexion it is taken as radial flow. Otherwise, the inflexion is attributed to this regime. Early time lasts until radial
flow. A possible double porosity transition is sought after radial flow. After it (or after radial flow) boundary effects are sought.

- **MODEL4** (Figure 3)
  Everything is early time, wellbore storage and skin.

- **MODEL5** (Figure 3)
  If there is a step, it is taken as radial flow. Everything before radial flow is early time. Boundary effects are sought if radial flow was found.

In the diagnosis of boundary effects, when a first stabilization is found, and if the behavior was homogeneous, the level of this stabilization is compared with the one corresponding to radial flow. When the difference is less than or equal to 0.25, the program considers that the response exhibits transient double porosity behavior.

The procedure described in this section enables us to choose an analytical model which will produce a similar response to the data. For the interpretation to be complete, we now need to actually match the two responses by adjusting the parameters inherent to the model. An automated type-curve matching will be used for this task provided that we have a first estimate of the parameter values. Section 4 describes the method used to arrive at the first estimates for some cases already developed.
4. QUANTITATIVE DESCRIPTION

A model is chosen to interpret a given section of the response because it can produce a shape similar to the one observed. When the parameters of this model vary, this shape will still be present but it may have a different amplitude and occur at a different time. If we have available relations between the parameters and the size of the specific shapes produced, we will have a way to give a first match, using the quantitative information present on the response. This is the reason why we have carried such information as the location of a hump, the length of a step in the representation of a given response.

Correlations are presented in this section for wellbore storage and skin and pseudosteady state double porosity.

1. Wellbore storage and skin

When a hump is found at early time, it indicates wellbore storage and skin.

- The type-curves for this model show that the height of the hump (when it is present) is a monotonic function of \( C_D \cdot e^{2S} \). This relation is shown on Figure 8 where \( \log(C_D \cdot e^{2S}) \) is plotted against \( \log\left((t_{D_{maximum}}/0.5)\right) \).

If radial flow is present on the response, we can get the value of \( C_D \cdot e^{2S} \) using this relation. Recognizing radial flow also enables us to find the value of the pressure match:

\[
\frac{\Delta p}{p_D} = \frac{\text{(pressure derivative)}_{\text{radial flow}}}{0.5}
\]

The time at which the maximum occurs is also a monotonic function of \( C_D \cdot e^{2S} \). This relation is shown on Figure 9 where \( \log(t_{D_{maximum}}) \) is plotted against \( \log(C_D \cdot e^{2S}) \).

Knowing \( C_D \cdot e^{2S} \) from the previous relation, we can get the time match:

\[
\frac{\Delta t}{t_D} = \frac{\Delta t_{maximum}}{t_{D_{maximum}}}
\]
2. **Pseudosteady state double porosity**

The following equation relates the value of the pressure derivative at the minimum to the transition to the value of the pressure derivative:

\[
(t_D \frac{dp_D}{dt_D})_{\text{minimum}} = \frac{1}{2} \left(1 + \omega^{1/(1-\omega)} - \omega^{\omega/(1-\omega)}\right)
\]  

This relation is shown on Figure 10.

Assuming that the procedure described in the previous paragraph has been applied, with the value of the pressure match we can get the value of \(t_D \frac{dp_D}{dt_D}\) at the minimum and therefore the value of \(\omega\).

With \(\omega\) and \(C_D e^{2S}\), we can get \((C_D e^{2S})_{f+m}\) and \(\lambda e^{-2S}\) with the following relations:

\[
\omega = \frac{(C_D e^{2S})_{f+m}}{(C_D e^{2S})_{f}}
\]  

\[
t_{D\text{minimum}} = (\omega / \lambda e^{-2S}(C_D e^{2S})_{f+m}) \ln \frac{1}{\omega}
\]

The relations presented in this section have been added to the program in table forms. Simple linear interpolation is used to compute the value of the parameters. This procedure will be illustrated with two examples in section 5.
5. APPLICATIONS

In this section, the program described is applied to two different examples generated using the model for pseudosteady state double porosity in an infinite medium. For both cases, the different steps followed during the execution are listed and the identification made by the program checked against the original data.

Example 1. (Figure 11)

The curve was generated using the following values:

$$(C_D e^{25})_{f+m} = 1.12 \times 10^7 \quad \omega = 0.0873 \quad \lambda e^{-15} = 1.35 \times 10^{-12}$$

First, the data are replaced by the symbolic representation given on Figure 12. This representation is used as input to the initial tree used in the interpretation. Figure 12 shows the path followed in this tree.

- NODE NO:
  - An inflexion is found on the curve which can accompany any of the following effects: wellbore storage and skin (damaged well), heterogeneous behavior, boundary effects.

- NODE N1:
  - A maximum is found before the inflexion: the response exhibits a hump which is due to wellbore storage and skin.

- NODE N3:
  - The shape immediately following the hump is not a minimum. It can either be a flat inflexion or a step. This shape will be taken to be radial flow. Node N6 is reached. It is a terminal node invoking the function MODEL2.

MODEL2:

The shape following the hump is taken to be radial flow. After radial flow a minimum is sought. This minimum is found and is indicative of pseudosteady state double porosity (transient double porosity would not produce any shape between the hump and the minimum).

- The inflexion found at this step is always the earliest one
Immediately after the trough, a stabilization is found which corresponds to radial flow regime for the total system (fissures + matrices).

The shapes remaining after the stabilization are input to the tree for boundary effects. There is actually no remaining shape: the reservoir is infinite.

In appendix A is given the output of the program for this example. For the interpretation model given, the value of the parameters has been calculated using the procedure described in section 4.

\[
(C_D e^{2S})_{f+m} = 0.924 \times 10^7 \quad w = 0.1 \quad \lambda e^{2S} = 1.24 \times 10^{-12}
\]

The response obtained with the estimated parameter values is plotted with the original data on Figure 13.

**Example 2.** (Figure 14)

The curve was generated using the following values:

\[
(C_D e^{2S})_{f+m} = 1.12 \times 10^7 \quad w = 0.0873 \quad \lambda e^{2S} = 1.35 \times 10^{-10}
\]

From example 1, only the value of \(X\) has been changed.

The data are first replaced by the symbolic representation shown on Figure 15. This representation is then input to the original tree. The path followed in this tree is shown on Figure 15.

- **Node NO:**
  An inflexion is found on the curve which can accompany any of the following effects: wellbore storage and skin (damaged well), heterogeneous behavior, boundary effects.

- **Node N1:**
  A maximum is found before the inflexion: the response exhibits a hump which is due to wellbore storage and skin.

- **Node N3:**
  A minimum immediately follows the hump: this minimum is either the bottom of a double porosity transition or radial flow followed by boundary effects. In the case of pseudo steady state double porosity, the transition starts before the end of wellbore storage. Otherwise, there would be at least a flat inflexion between the hump and the minimum.

The two possibilities are regrouped under MODEL1.
**MODEL 1:**

There is an inflexion following the minimum: it indicates that the curve tends towards a stabilization. The level of this stabilization is taken to be the level of the next shape after the inflexion (symbolically represented by a "STEP").

If the difference between the levels of the stabilization and the minimum is lower than or equal to 0.25, transient double porosity will be considered as the only interpretation model. This is not the case here.

Two consecutive interpretations are performed.

1. **First case**
   - The minimum is considered as the bottom of a pseudosteady state double porosity transition.
   - The step at the end of the data is taken to be radial flow of the total system.
   - The remaining shapes are input to the tree for boundary effects. There is actually no remaining shape. The reservoir is infinite.

2. **Second case**
   - The minimum is considered as radial flow.
   - The remaining shapes are input to the tree for boundary effects.
   - The path followed in this tree is shown on Figure 15.

- **NODE B1:**
  - An upward inflexion is found which indicates the effect of one or several sealing faults.

- **NODE B3:**
  - After the previous transition, the curve exhibits a stabilization.

- **NODE B5:**
  - No other shape remains: the interpretation is complete.

In appendix B is given the output of the program for this example. For both interpretations performed, the relations presented in section 4 were used in order to get the value of some parameters associated with the models. The values found for the double porosity case are:

\[
(CDe^{25})_{f+m} = 1.04 \times 10^7 \quad w = 0.13 \quad \lambda e^{-25} = 0.86 \times 10^{-10}
\]

On figure 16, the double porosity response proposed by the program is plotted with the original data.
Discussion

The match obtained in the first example would probably not be significantly improved by automated type-curve matching. This is what we expect from the application of the program to theoretical curves.

In the second example, two identification models were proposed which can both produce a similar response to the original one— at a qualitative level of representation. Besides, when the minimum is considered as radial flow followed by boundary effects, the level of the stabilization is 1.15. The shapes diagnosed as boundary effects would therefore be close to the effect of one single sealing fault. However, in this case $C_D e^{45}$ is on the order of $10^{50}$. With such a value, it is likely that the trough would not be so sharp.

This example suggests the following:

1. Quantitative information present on the response can in certain cases be used in the identification of the model.

2. It is not possible to gather at once all the knowledge involved in the interpretation. The best way to validate the program is to check its behavior on a wide range of examples against that of an expert and modify it if necessary.

The values of the parameters found by the program for example 2 differ slightly from the actual ones. This difference is emphasized on Figure 16 where the response obtained is plotted along with the original data.

The reason for this difference is that the double porosity transition occurs before the end of wellbore storage. Because these two regimes influence each other, the characteristics of the hump and the trough are altered. We can estimate this alteration by comparing the original data for example 1 and example 2 since only $X$ varies from one to the other.

The two original responses are plotted on Figure 17. We can see on this plot that the location of the maximum is not affected. This is the reason why the curve obtained with the program in example 2 matches the hump (Figure 16).

On Figure 18, one of the curves was translated in order to compare the locations of the minimum. Since it is lower in example 2, the program has overestimated $w$ as we can verify from the output in appendix B. Since the trough becomes deeper as $w$ becomes smaller, this causes the curve obtained by the program to be above the original one (Figure 16).

We must keep in mind that the function of the program is not to provide a
definitive and optimal match. Its objective is simply to choose an appropriate model (or models) and provide a first estimate to be used subsequently in an automated type-curve matching. The match obtained in both examples is certainly close enough for this purpose.

For the second example, we noticed that the consideration of quantitative information would have enabled us to select only one interpretation model. Automated type-curve matching provides a way to try several interpretation models very quickly. It might therefore be interesting to consider anyway all the admissible models at a qualitative level of representation, and decide which one is the most appropriate by comparing the matches obtained.
6. CONCLUSION

Using an Artificial Intelligence approach, it is possible to program a computer to perform model identification on smoothed data. AI language and techniques provide proper ways of representing the response symbolically and reproducing the reasoning processes involved in the interpretation. Tree structures are convenient to organize the knowledge of the various analytical responses, and can be easily augmented with new models.

Simple correlations between the parameters associated with a particular model and the characteristics of the shapes it produces can be used to give a first match from which an automated type-curve matching can proceed. Such correlations were presented for wellbore storage and skin and pseudosteady state double porosity. Because they do not take into account the possible overlap of different regimes, these correlations can sometimes be inaccurate. They are however accurate enough for the purpose of giving a first match. Other, relations need to be developed for other regimes.

For the interpretation to be fully automated, a smoothing algorithm must be developed which will be used on top of the program described in this report. The preparation of the data needs to be more than a numerical procedure. It must integrate knowledge about the producible curves in order to eliminate shapes which could remain on the response after a simple smoothing and not be representative of any phenomenon in the reservoir.

Finally, it is not possible to represent exhaustively at once the knowledge required for the interpretation. To make this knowledge as complete as possible, a verification phase is necessary: the analysis performed by the program should be checked on a wide range of tests against that of an expert. If necessary, modifications should be made to the program so that both interpretations match.
Figure 1.
Figure 2.

INITIAL TREE
Figure 3.

MODEL5

\[ t_D \frac{dP_D}{dt_D} \]

MODEL4

\[ t_D \frac{dP_D}{dt_D} \]

max

step

step
Figure 4.

**MODEL3**

\[ t_D \frac{dP_D}{dt_D} \]

\[ t_D \frac{dP_D}{dt_D} \]

\[ t_D \frac{dP_D}{dt_D} \]
Figure 5

MODEL2

\[ t_D \frac{dP_D}{dt_D} \text{ max} \]

MODEL1

\[ t_D \frac{dP_D}{dt_D} \text{ max} \]

\[ t_D \text{ step} \]

\[ t_D \text{ step} \]

\[ t_D \text{ inf} \]
**Figure 6**

Tree for boundary effects

- **IU**: Inflexion up
- **UT**: Upward trend
- **DT**: Downward trend
- **IF**: Flat inflexion
- **MAX**: Maximum
- **STEP**: Stabilization

![Diagram](image)
Figure 7.

Examples of boundary effects.
Figure 8.

Wellbore storage and skin, infinite homogenous reservoir.

\[ \log(\frac{dP}{dt}/dP_{\text{max}}) \]
Double porosity pseudo steady state.

Figure 10.
Figure 11.
Figure 12.

Example 1.

Representation

((MAXIMUM (1.0 . 0.6)) (INFLEXION (1.7 . 0.2) DOWN) (INFLEXION (2.6 . -0.3) FLAT) (INFLEXION (3.8 . -0.7) DOWN)
  (MINIMUM (4.3 . -0.8)) (INFLEXION (4.5 . -0.7) UP) (STEP 10E-5 (5.6 . -0.3) (6.8 . -0.3)))

MODEL 1
MODEL 2
MODEL 3
MODEL 4
MODEL 5
Figure 15.
Example 2.

Representation

((MAXIMUM (-1.0 . 0.6)) (INFLEXION (1.8 . 0.2) DOWN) (MINIMUM (2.4 -.7) (INFLEXION (2.5 -.6) UP) (STEP 10E-5 (3.6 -.3) (6.8 -.3)))

MODEL 3
MODEL 4
MODEL 5

38
Figure 16.
Figure 17.
Bibliography


Appendix A.

NODE N1:

An inflexion has been found on the curve which could accompany any of the following effects:
- End of Wellbore Storage.
- Heterogeneous behavior of the reservoir.
- Boundary effects.

NODE N3

The inflexion found previously is preceded by a maximum: the derivative exhibits a hump which is due to wellbore storage and skin (damaged well) unless transient double porosity is diagnosed later.

NODE N6

Directly after the hump on the early time section of the response is a minimum:
- This minimum is radial flow; it is followed by boundary effects.
  or
- This minimum is the bottom of a double porosity transition. In the case of a pseudosteady state double porosity, the transition starts before the radial flow of the fissures is developed.

Early time is wellbore storage and skin.

A transition due to heterogeneous behavior was found:
- Double porosity.
  or
- Double permeability.
Since the response exhibit radial flow for the fissures, it cannot be transient double porosity.

The reservoir is infinite.

\[(CDe2s) f = 9.24061920F7\]

\begin{align*}
time \text{ match} & = 1.00095712 \\
pressure \text{ match} & = 1.00047659 \\
\omega & = 0.100972906 \\
\Lambda & = 1.24345533F^{-12}
\end{align*}
NODE N1:
An inflexion has been found on the curve which could accompany any of the following effects:
- End of Wellbore Storage.
- Heterogeneous behavior of the reservoir.
- Boundary effects.

NODE N3
The inflexion found previously is preceded by a maximum: the derivative exhibits a hump which is due to wellbore storage and skin (damaged well) [unless transient double porosity is diagnosed later].

NODE N6
Directly after the hump on the early time section of the response is a minimum:
- This minimum is radial flow; it is followed by boundary effects.
  or
- This minimum is the bottom of a double porosity transition. In the case of a pseudosteady state double porosity, the transition starts before the radial flow of the fissures is developed.

*********************** FIRST CASE:
The minimum is considered as bottom of a double porosity transition.

Pseudosteady state double porosity.
Early time is CDe2S corresponding to the fissures.
The reservoir is infinite.

\[(\text{CDe2S})_f = 7.93654560F7\]
\[\text{time match} = 1.00045374\]
\[\text{pressure match} = 0.997486352\]

\[\Omega = 0.130815416\]
\[\Lambda = 8.53287913F-11\]

*********************** SECOND CASE:
The minimum is considered as radial flow followed by boundary effects.

Early time is wellbore storage and skin.

The reservoir is homogeneous unless transient double porosity diagnosed later.

NODE B1
And now boundary effects are analyzed.

NODE B3
An upward inflexion was found on the curve which is characteristic of one (or several sealing) fault(s) [can also be due to transient double porosity if no heterogeneity has been found yet].
The previous effect is due to boundaries.

NODE B5
After the inflexion previously diagnosed, the curve stabilizes

All the test has been analyzed.

\[ CDe2s = 1.25084143F24 \]
\[ \text{time match} = 3.13263607 \]
\[ \text{pressure match} = 2.7876091 \]
Appendix C.

WELL TEST INTERPRETATION

Global Variables

(defvar *NUMBER-POINTS* 6)
(defvar *EPSILON* 0.01)
(defvar *EPSILON-SLOPE* 0.01)
(defvar *INF* (1))
(defvar *MAX* ())
(defvar *MAX1* ())
(defvar *MIN* ())
(defvar *INTER* nil)
(defvar *SEGMENTS* nil)
(defvar *MAX-WBS* nil)
(defvar *IN-ET* nil)
(defvar *IARF-AT* nil)
(defvar *RF-B1* nil)
(defvar *RF-B2* nil)
(defvar *MIN-DPT* nil)
(defvar *MIN-DPP* nil)
(defvar *MAX-BETA* nil)
(defvar "DEWIN-OMEGA-TABLE")
(defvar PD-CDE2S-TABLE'
  '(0.017 . 0.0)
   (0.016 . 0.477121)
   (0.22 . 1.0)
   (0.398 . 2.0)
   (0.531 . 3.0)
   (0.633 . 4.0)
   (0.778 . 6.0)
   (0.892 . 8.0)
   (0.98 . 10.0)
   (1.146 . 15.0)
   (1.265 . 20.0)
   (1.43 . 30.0)
   (1.55 . 40.0)
   (1.64 . 50.0)
   (1.72 . 60.0))
(defvar *CDE2S-TD-TABLE'
  '(0. 0.20)
    (0.477121 6.5)
    (1. 4.7)
    (2. 5.2)
    (3. 6.0)
    (4. 7.2)
    (6. 9.5)
    (8. 12)
    (10. 14.5)
    (15. 20.5)
    (20. 26.0)
    (30. 39.0)
    (40. 50.0)
    (50. 64.0)
    (60. 76.0))
(defvar *CDE2S* 0)
(defvar *PMATCH* 0)
(defvar *TMATCH* 0)
(load "DERMIN.LISP")
(load "GLOBAL_REGIMES.LISP")
(setf *DERMIN-OMEGA-TABLE*
  '((-9.67936  -11.5129)
   (-9.04325  -10.8198)
   (-8.67274  -10.4143)
   (-8.41062  -10.1266)
   (-8.20778  -9.90349)
   (-8.04236  -9.72117)
   (-7.90273  -9.56702)
   (-7.78195  -9.43348)
   (-7.67556  -9.31570)
   (-7.58050  -9.21034)
   (-6.95800  -8.51719)
   (-6.59640  -8.11733)
   (-6.34110  -7.82405)
   (-6.14386  -7.60090)
   (-5.98325  -7.41858)
   (-5.84785  -7.26443)
   (-5.73088  -7.13090)
   (-5.62795  -7.01312)
   (-5.53608  -6.90776)
   (-4.93700  -6.21461)
   (-4.59146  -5.80914)
   (-4.34887  -5.52146)
   (-4.16234  -5.29832)
   (-4.01110  -5.11600)
   (-3.88410  -4.96184)
   (-3.77478  -4.82331)
   (-3.67891  -4.71053)
   (-3.59362  -4.60517)
   (-3.04494  -3.91202)
   (-2.73573  -3.50656)
   (-2.52264  -3.21888)
   (-2.36141  -2.99573)
   (-2.23252  -2.81341)
   (-2.12569  -2.65926)
   (-2.03482  -2.52573)
   (-1.95601  -2.40795)
   (-1.88663  -2.30259)
   (-1.45885  -1.60944)
   (-1.23415  -1.20337)
   (-1.08727  -0.916291)
   (-0.980829  -0.693147)
   (-0.898823  -0.510826)
   (-0.833009  -0.356675)
   (-0.778618  -0.223144)
   (-0.732660  -0.105361)))
(setf *IARF-ET* nil)
(setf *IARF-AT* nil)
(setf *MIN-DPP* nil)
(setf *MIN-DPT* nil)
(setf *RF-BE1* nil)
(setf *RF-BE2* nil)
(setf *MAX-WBS* nil)
Macros used in the functions

(defun node
  (ancestor
  successors
  test
  result - test
  model)

  (defnode N 'ancestor 'successors 'test 'test - result 'model)

(defun defnode (name ancestor successors test result - test model)
  (setf ,name (make - node
    :ancestor ,ancestor
    :successors ,successors
    :test ,test
    :result - test ,result - test
    :model ,model)))

(defun bnode
  test
  model)

  (defnode bnode (name test model)
  (setf ,name (make - bnode
    :test ,test
    :model ,model)))

(defun get-slope segment)
  (car segment)
(defmacro GET-SLOPE (segment)
  `(car ,segment))

;; (get-length segments)
;; ==> (cadr segment)

(defmacro GET-LENGTH (segment)
  `(cadr ,segment))

;; (get-start segment)
;; ==> (caddr segment)

(defmacro GET-START (segment)
  `(caddr ,segment))

;; (get-end segment)
;; ==> (caddr segment)

(defmacro GET-END (segment)
  `(caddr ,segment))

(defmacro END-POINT (straigth)
  `(cdar (cdddr ,straigth)))

;; The following function recognizes a final upward trend.

(defmacro INFLEXION-UP (shape)
  `(and
    (eq (car ,shape) 'INFLEXION)
    (eq (caddr ,shape) 'UP)))

(defmacro INFLEXION-DOWN (shape)
  `(and
    (eq (car ,shape) 'INFLEXION)
    (eq (caddr ,shape) 'DOWN)))

(defmacro INFIXXION-FLAT (shape)
  `(and
    (eq (car ,shape) 'INFLEXION)
    (eq (caddr ,shape) 'FLAT))
;; This first function reads the original string of data and builds a list of pairs
;; with them.
;; (PAIR data) with data = "1.0 2.0 3.0 4.0"
;; => ((1.0 . 2.0) (3.0 . 4.0))

(defun PAIR (filename)
  (let* ((stream (open filename :direction :input))
         (time1 (read stream 't))
         (pressure1 (read stream 't)))
    (do ((time (read stream nil 'EOF)
            (read stream nil 'EOF))
         (pressure (read stream nil 'EOF)
                   (read stream nil 'EOF))
         (pairs `((,time1 ,pressure1))
                    (append pairs new-pair)))
      ((eq pressure 'EOF)
        (cond ((eq time 'EOF)
                   (return pairs)
                   (close stream))
              (t (format t "-*** Data must be in even number ***")))))
  (setf new-pair `((,time . 'pressure))))))

;; Each dotted pair in the list represents a point. The following functions
;; transforms it in a list of segments binding two consecutive points.
;; The segment starting at (1.0 . 2.0) and ending at (3.0 . 6.0) has the
;; form (2.0 2 (1.0 . 2.0) (3.0 . 6.0))
;; where
;; - 2.0 is the slope.
;; - 2 the number of point.
;; - (1.0 . 2.0) the starting point.
;; - (3.0 . 6.0) the ending point.
;; The general form of a segment is thus
;; (slope length start-point end-point)
;; start-point and end-point being dotted pairs.

(defun SEGMENTATION (list-of-points)
  (let ((start-point (car list-of-points))
         (end-point (cadr list-of-points)))
    (cond ((null end-point) nil)
          (t (cons
              (make-segment start-point
                            end-point)
              (segmentation (cdr list-of-points)))))))
(defun MAKE-SEGMENT (start-point end-point)
  (let* ((time1 (car start-point))
          (pressure1 (cdr start-point))
          (time2 (car end-point))
          (pressure2 (cdr end-point))
          (slope (/ (- pressure2 pressure1)
                     (- time2 time1))))
    (list slope '2 start-point end-point)))

;; Some portions of the curve may be straight lines. In such portions, we want to
;; have one segment only. The following functions does this.
;; When two consecutive segments have slopes which differ by less than *epsilon*,
;; they are replaced by a new one such as
;; -The slope is the average of the slopes.
;; -The length is the sum of the lengths \(-1\) (number of points).
;; -The starting point is the starting point of the first segment.
;; -The end point is the end point of the second segment.

(defun EXTEND (list-of-segments)
  (let ((first-segment (car list-of-segments))
          (second-segment (cadr list-of-segments)))
    (cond
      ((null second-segment) (cons first-segment 0))
      ((abs (- (get-slope first-segment)
                (get-slope second-segment)))
        *epsilon*) ;The two segments will be replaced
                 ;by a single one.
        (extend (cons
                  (prolong first-segment second-segment)
                  (cadr list-of-segments)))))
    (t (cons first-segment
      (extend (cdr list-of-segments)))))))

;; PROLONG is the function which actually makes a new segment out of two which meet the
;; condition on the slopes.

(defun PROLONG (segment1 segment2)
  (let '((start-point (get-start segment1))
          (end-point (get-end segment2))
          (slope (+ (get-slope segment1)
                    (get-slope segment2)))
          (2.0))
    (length (+ (get-length segment1)
               (get-length segment2)
               \(-1\))))
    (list slope length start-point end-point)))

;; Once the segments have been extended (according to the chosen *epsilon*) the list
;; is transformed into a list of characteristic features along with their attributes.
;; - Straight lines : Segments which length is greater than *number-points*
;; - They are STEP if the slope is less than *epsilon-slope*
;; - represented by (slope start-point end-point)
;; - Extrema : Maxima or minima
;; - represented by (name point) [name being maximum or minimum]
;; - Inflexion : represented by (inflexion point)
(defun REPRESENTATION (list-of-segments)
  (let ((new-rep ()))
    (do ((remain-segments (cdr list-of-segments) (cdr remain-segments))
         (segment (car list-of-segments) (car remain-segments))
         (representation () (append representation new-rep))
         (number (1+ number))
         (ante-slope () slope) ;slope of the previous segment
         (slope (get-slope (car list-of-segments)) post-slope)
         (post-slope (get-slope (cadr list-of-segments)) ;slope of the next one
                    (get-slope (cadr remain-segments))))
      ;; end form of the do
      
      (if (null segment)
          (return representation))
      
      (let ((start (get-start segment))
            (end (get-end segment)))
        (cond ;; straight line
          ((>= (get-length segment) *number-points*)
           (setf *inter* (append *inter* ',number)))
          (if (< (abs (get-slope segment)) *epsilon-slope*) ;flat section case
            (setf new-rep '((STEP ,slope ,start ,end)))
            (setf new-rep '((STRAIGHT ,slope ,start ,end))))
            (or (null ante-slope) ;The first and last segments cannot
                (null post-slope)) ;be extrema or inflexion.
            (setf new-rep nil))
            ;; the two following cases are for maxima. The first one for a "flat"
            ;; maximum, the second one for a sharp one.
            
            ((and (> ante-slope 0.0)
                  (< post-slope 0.0)
                  (= ante-slope 0.0)
                  (< slope 0.0))
              (progn
                (setf *inter* (append *inter* ',number))
                (setf point (cons
                              (mean (car start) (car end))
                              (mean (cdr start) (cdr end))))
                (setf new-rep '((MAXIMUM ,point))))
            (and (< ante-slope 0.0)
                 (< slope 0.0))
            (setf *inter* (append *inter* ',number))
            (setf new-rep '((MAXIMUM ,start))))
            ;; the two following cases are for minima. The first one for a "flat"
            ;; minimum, the second one for a sharp one.
            
            ((and (< ante-slope 0.0)
                  (> post-slope 0.0)
                  (= ante-slope 0.0)
                  (> slope 0.0))
              (progn
                (setf *inter* (append *inter* ',number))
                (setf point (cons
                              (mean (car start) (car end))
                              (mean (cdr start) (cdr end))))
                (setf new-rep '((MINIMUM ,point))))
            (and (<= ante-slope 0.0)
                 (> slope 0.0))
            (setf *inter* (append *inter* ',number))
            (setf new-rep '((MINIMUM ,end))))
      )
)

54
(defun mean (value1 value2) 
  (/ (+ value1 value2) 2.0))

;; MEMBER-SHAPE acts like MEMBER on the list of shapes. Given a
;; TYPE OF SHAPE, for instance `INFLEXION, it outputs either nil
;; if there is no inflexion, or a list which first element is the
;; first occurrence found and the last element is shape2.
;; MEMBER-SHAPE searches inclusively between shape1 and shape2.
;; If there are not supplied they default to the first and last
;; elements in the list of shapes

(defun MEMBER-SHAPE (shape shapes &optional (shape1 (car shapes))
  (shape2 (car (last shapes))))
  (if (null shape1) (setf shape1 (car shapes)))
  (cond
    ((null (member shape2 shapes :test #'equal)) nil)
    (t
      (cond
        ((eq (caar shapes) shape)
         (cond
          ((member shape1 (cdr shapes) :test #'equal)
           (member-shape shape (cdr shapes) shape1 shape2))
          (t (MEMBER-SHAPE shape (cdr shapes) shape1 shape2))))
        (t (MEMBER-SHAPE shape (cdr shapes) shape2))))))

;; REM-FROM removes in shapes all the elements from shape1 to shape2.
;; When shape1 is not provided, starts removing at the beginning.
;; When shape2 is not provided, removes down to the end.

(defun REM-FROM (shapes &optional (shape1 (car shapes)) (shape2 (car (last shapes))))
  (if (null shape1) (setf shape1 (car shapes)))
  (cond
    ((not (member shape2 shapes :test #'equal)) shapes)
    ((eq (caar shapes) shape1)
     (rem-from (cdr shapes) (cdr shapes) shape2))
    (t (cons (car shapes) (rem-from (cdr shapes) shape1 shape2))))))

;; NEXT-SHAPE gives the successor of a given shape in shapes.

(defun NEXT-SHAPE (shape shapes)
  (cadr (member shape shapes :test #'equal)))

;; HEIGHT gives the level of a given shape. For a segment, it takes the
;; average of start and end point.

(defun height (shape)
(cond
    ((or (eq (car shape) 'INFLEXION)
            (eq (car shape) 'MINIMUM)
            (eq (car shape) 'MAXIMUM))
      (cadr shape))
    (t
      (mean (cadr (get-end shape))
            (cadr (get-start shape)))))))
;; The last expression works for normal segments as well as STEP or STRAIGHT
;; Once the list of shapes has been created, if the last one does not
;; correspond to the last segment, we want to define a final trend.
(defun complete (shapes)
  (let ((n (car (last *INTER*)))
        (m (length *segments*))
    (cond ((eq n m) shapes)
      (t (do* ((iter n (+ 1 iter))
                (slope (get-slope (nth n *segments*))
                        (+ slope (get-slope (nth iter *segments*)))))
           ;; Initially, iter=n. nth gives the element of index iter+1 with index 1
           ;; for the first segment. We stop when iter+1 is equal to m.
           ;; We take the average of the slope for (m-n) points.
           (when (= (+ 1 iter) m) (setf slope (/ slope (- m n))))
           (cond ( (> slope *epsilon-slope*)
                   (setf shapes (append shapes '((UTREND ,slope))))
           (cond ( (> (abs slope) *epsilon-slope*)
                   (setf shapes (append shapes '((DTREND ,slope))))))
                (return shapes)
     ))))))
;; The following function is used in the recognition of boundary effects.
(defun UPWARD (shape)
  (cond ((eq (car shape) 'UTREND) (setf *UT* shape))
        ((and (eq (car shape) 'STRAIGHT)
              (> (get-slope (shape) 0.)
               (setf *UT* shape))
            (t nil)))
  (defun DOWNWARD (shape)
  (or (eq (car shape) 'DTREND)
      (inflexion-down shape)
      (and (eq (car shape) 'STRAIGHT)
           (< (get-slope shape) 0.))))))
### WELL TEST ANALYSIS

Network For The Interpretation

---

The creation of the Network used for the interpretation is done here

```lisp
(defun FNO (shapes)
  (setf *INF* (car (member-shape 'INFLEXION shapes))))

(defun FN1 (shapes)
  (setf *MAX* (car (member-shape 'MAXIMUM shapes () *INF*))))

(defun FN2 (shapes)
  (setf *MAX1* (car (member-shape 'MAXIMUM shapes))))

(defun FN3 (shapes)
  (let ((next (next-shape *INF* shapes)))
    (if (eq (car next) 'MINIMUM)
        (setf *MIN* next))))

(defun change-node (node shapes)
  (funcall (node-test node) shapes)
  (if (null (eval (node-result-test node)))
      (eval (cadr (node-successors node))))
  (eval (car (node-successors node))))

(defun interpretor (node shapes)
  (if (not (equal node NO))
      (funcall (node-model node))
    (cond
      ((null (node-successors node)) (funcall (node-test node) shapes))
      (t (interpretor (change-node node shapes) shapes))))
```

And now we define the function interpretor which uses "change-node" to perform the interpretation. When it is called, "node" is the first node of the tree: NO0

Definitions of the functions associated to each model

57
An inflexion has been found on the curve which could accompany any of the following effects:
- End of Wellbore Storage.
- Heterogeneous behavior of the reservoir.
- Boundary effects.

Since no inflexion has been found on the pressure derivative, the behavior is HOMOGENEOUS. The only late-time behavior that could occur without creating the inflexion on the pressure derivative are the following:
- Pseudo Steady State flow [Closed System] (slope 1)
- Linear Flow [parallel boundaries] (slope 1/2)
- Steady State flow [Constant Pressure or closed reservoir if drawdown] (Drop on the derivative)

The inflexion found previously is preceded by a maximum:
the derivative exhibits a hump which is due to wellbore storage and skin (damaged well) [unless transient double porosity is diagnozed later].

A maximum is present on the curve which corresponds to wellbore storage. Since no inflexion had been found on the response, the test must have been stopped before radial flow is reached.

There is no maximum on the curve: the early time part can either be a low CD2S (no hump) or a fracture. The test ends during or before radial flow is reached.

Directly after the hump on the early time section of the response is a minimum:
- This minimum is radial flow; it is followed by boundary effects.
or
- This minimum is the bottom of a double porosity transition. In the case of a pseudo steady state double porosity, the transition starts before the radial flow of the fissures is developped.

The shape following the maximum is not a minimum. Therefore, the first solution proposed at the previous step will be chosen: Wellbore storage and skin - High CD2S.

Since there is no maximum on the curve, the early time section cannot be a high CD2S. We will have to decide between low CD2S or fracture. The presence of the inflexion previously diagnozed indicates that either heterogeneities or boundary effects are present.

Next is the definition of the network used for boundary effects
Node used in this case are different from those used before. They are defined in the file structure.
The functions to move inside the tree are also different

(defun change-bnode (bnodes shapes)
  (funcall (bnodes-test bnodes) shapes))

(defun binterpretor (bnodes shapes)
  (cond
    ((and (equal bnodes B1)
           (null (funcall (bnodes-test bnodes) shapes)))
     (format stream "\(-\)\-
The reservoir is infinite.")
    ((null bnodes)
     (format stream "\(-\)\- All the test has been analyzed.")
     ((null shapes) (funcall (bnodes-model bnodes))
      (format stream "\(-\)\- All the test has been analyzed.")
      (t (funcall (bnodes-model bnodes)
        (binterpretor (change-bnode bnodes shapes) (cdr shapes)))))))

;; The tree for searching boundary effects walks down to the end of the
;; list. Shapes in the portion recognized as boundary effects are taken
;; one by one, analyzed, removed. This is the reason why interpretor
;; recurs with (cdr shapes)

(defun LATE-TIME (shapes)
  (binterpretor B1 shapes))

(defun FB1 (shapes)
  (let ((next (car shapes)))
    (cond
      ((null next) nil)
      ((inflexion-up next) B3)
      ((upward next) B2)
      ((downward next) B4)
      (t (format t "\(-\)\- Something strange\ldots\ldots"))))

(defun FB3 (shapes)
  (let ((next (car shapes)))
    (cond
      ((null next) nil)
      (or (eq (car next) 'inflexion) (eq (car next) 'step)
       (cond
         ((null *RF-BE1*)
          (setf *RF-BE1* next)
          (RULE1)
          (t (setf *RF-BE2* next)))
         B5)
      (t (format t "\(-\)\- Something strange\ldots\ldots"))))

(defun FB5 (shapes)
  (let ((next (car shapes)))
    (cond
      ((null next) nil)
      ((upward next) B2)
      ((inflexion-up next) B3)
      ((downward next) B4)
      (t (format t "\(-\)\- Something strange\ldots\ldots"))))
RULE1 is used to differentiate transient double porosity from boundary effects. This rule is used at the moment we assign a value to *RF-BEl*.

(defun RULE1 ()
  (cond
    ((or (not (null *MIN-DPT*)) (not (null *MIN-DPP*))) nil)
    ((< (- (height *RF-BEl*) (height *IARF-ET*)) (+ (log 2 10) *epsilon*))
      (format stream "-S" *RF-BEl*)
      (format stream "-S" *IARF-ET*)
      (format stream ":-%%")
      The response exhibits transient double porosity behavior.
    )
    (setf *MAX-BETA* *MAX-WBS*)
    (setf *MAX-WBS* NIL)
    (setf *IARF-AT* *RF-BEl*)
    (setf *RF-BEl* nil)
    (setf *IARF-ET* nil))

(t (format stream "-S")
The previous effect is due to boundaries.
))

;;; Function associated with each node of the B-tree

(defun BMOD1 ()
  (format stream "-%-% NODE B1
And now boundary effects are analyzed.
")

(defun BMOD2 ()
  (cond
    ((eq STRAIGHT (car *UT*))
      (if (< (abs (- (get-slope *UT*) 1))
          (abs (- (get-slope *UT*) 0.5))
          (format stream "-%-% NODE B2
An upward trend was found on the curve which is due to Pseudo Steady State (closed reservoir).
"))
        (format stream "-%-% NODE B2
An upward trend was found on the curve which corresponds to linear flow: effect of two parallel no flow boundaries.
"))
      (t (format stream "-%-% NODE B3
An upward trend appears on the end of the curve due to no flow boundaries.
"))))

(defun BMOD3 ()
  (format stream "-%-% NODE B3
An upward inflexion was found on the curve which is characteristic of one (or several sealing) fault(s) [can also be due to transient double porosity if no heterogeneity has been found yet].
")

(defun BMOD4 ()
  (format stream "-%-% NODE B4
A downward trend was found which is characteristic of a constant pressure boundary. No further analysis will be done despite the fact that no flow boundaries could also influence the response later.
")

(defun BMOD5 ()
  (format stream "-%-% NODE B5
After the inflexion previously diagnozed, the curve stabilizes.
")

(defun BMOD6 ()
  (format stream "-%-% NODE B6
The curve drops due to the effect of one (or several) constant pressure fault(s). No further analysis will be done.
")

60
(defun get-cde2s ()
  (let* ((p-iarf (if (null *IARF-AT*)
                   (height *IARF-ET*)
                   (height *IARF-AT*)))
         (t-max (caadr *MAX-WBS*))
         (p-max (cdadr *MAX-WBS*))
         (log-cde2s (lookup (- p-max p-iarf) *pd-cde2s-table*))
         (log-pmatch (- (log 0.5 10) p-iarf)))
    (setf *tmatch* (/ (lookup log-cde2s *cde2s-td-table*) (expt 10 t-max)))
    (setf *CDe2S* (expt 10 log-cde2s))
    (setf *pmatch* (expt 10 log-pmatch)))

(defun lookup (value table)
  (cond ((<= value (caar table)) (cdar table))
        ((> value (caar (last table))) (cdar (last table)))
        (t (do ((tab table (cdr tab))
                 (high (cadr table) (cadr tab))
                 (low (car table) high))
               ((and (< value (car high))
                     (>= value (car low)))
                (return (interpol value low high))))))

(defun interpol (value low high)
  ;; Both low and high are dotted pairs
  (let ((slope (/ (- (cdr high) (cdr low))
                 (- (car high) (car low))))
         (+ (cdr low) (* (- value (car low)) slope))))
;;; ** WELL TEST ANALYSIS **
;;; ** Regime Distinction **

(defun EARLY-TIME (shapes)
  (let ((slope (if (eq (caar shapes) 'STRAIGHT)
                   (get-slope (car shapes))
                   (get-slope (longest shapes))))
       (cond
         (<= (abs (- slope 0.5)) *epsilon*)
         (setf *ET* 'FRACTURE)
         (setf *ET* 'LCDe2S)
         (if (and (not (null *MAX*))
                  (> (height *MAX*) (height *IARF-AT*))
                  (setf *MAX-WBS* *MAX*)))))

(defun longest (shapes)
  (let ((longest-segment (car *segments*)
         (do ((remain-segments *segments* (cdr remain-segments))
             (count 1 (+ 1 count))
             (segment (car *segments*) (car remain-segments))
             (or (null remain-segments)
                 (eq count (car *inter*)))
             (return longest-segment))
      (if (> (get-length segment) (get-length longest-segment))
        (setf longest-segment segment))))

;; The following function is the one called in the case of MODEL 1 in
;; the main interpretation tree

(defun MODEL1 (shapes)
  (let ((next (next-shape *MIN* shapes))
        (cond
         (not (eq 'INFLEXION (car next)) (MODEL2 shapes))
         (let ((stabilization (next-shape next shapes))
               (if (null stabilization)
                   (setf stabilization (car (last *segments*)))
                   (cond
                    (<= (- (height stabilization) (height *MIN*)
                         (+ (log 2 10) *epsilon*))
                        (format stream "-%-%
                        (Setf *MAX-BETA* *MAX*)
                        (setf *MAX-WBS* nil)
                        (setf *MIN-DPT* *MIN*)
                        (setf *IARF-AT* stabilization)
                        (late-time (rem-from shapes nil *IARF-AT*)))
                    (t (format stream "-%-%
                        "Transient double porosity. Only this case will be developped."")
                        (setf *MAX-BETA* *MAX*)
                        (setf *MAX-WBS* nil)
                        (setf *MIN-DPT* *MIN*)
                        (setf *IARF-AT* stabilization)
                        (late-time (rem-from shapes nil *IARF-AT*))))
        (t (format stream "-%-%
                        "FIRST CASE:
                        The minimum is considered as bottom of a double porosity transition."")
                        (MODEL1-A shapes)
                        (load "global_regimes.lisp")
                        (format stream "-%-%
                        SECOND CASE:})

62
The minimum is considered as radial flow followed by boundary effects.

(DEFUN MODEL2- A (SHAPES) ; When the minimum is considered as bottom of a dp transition
  (SETF *MIN-DPP* *MIN*)
  (LET* ((INF (NEXT-SHAPE *MIN* SHAPES)) ; We know it is present
    (COND ((NULL (NEX-T-SHAPE INF SHAPES)))
      (COND ((NULL MIN) (FORMAT STREAM "-%-%
          "1)) ; The reservoir is homogeneous unless transient double porosity diagnozed later.
      (T (LET ((NEXT (NEXT-SHAPE MIN SHAPES))
          (LATE-TIME (RE-MM-FROM SHAPES NIL *IARF-AT*)))
          (GET-EARLY SHAPES)
          (GET-LATE)
          )))
  ))

;; MODEL2-A looks for a double porosity transition on the curve

(DEFUN MODEL2-A (SHAPES) ; The reservoir is homogeneous unless transient double porosity diagnozed later.
  (LET ((MIN (CAR (MEMBER-SHAPE 'MINIMUM SHAPES))))
    (COND ((NULL MIN) (FORMAT STREAM "-%-%
          "1)) ; A transition due to heterogeneous behavior was found:
      (T (LET ((NEXT (NEXT-SHAPE MIN SHAPES)))
          (FORMAT STREAM "-%-%
            "1)
          (GET-EARLY SHAPES)
          (GET-LATE)
          )))

;; MODEL2-A looks for a double porosity transition on the curve

(DEFUN MODEL2-A (SHAPES) ; The reservoir is homogeneous unless transient double porosity diagnozed later.
  (LET ((MIN (CAR (MEMBER-SHAPE 'MINIMUM SHAPES))))
    (COND ((NULL MIN) (FORMAT STREAM "-%-%
          "1)) ; A transition due to heterogeneous behavior was found:
      (T (LET ((NEXT (NEXT-SHAPE MIN SHAPES)))
          (FORMAT STREAM "-%-%
            "1)
          (GET-EARLY SHAPES)
          (GET-LATE)
          )))

;; MODEL3
(defun MODEL3 (shapes)
  (let ((step (car (member-shape shapes nil *INF*))))
    ;; Is there a step before the inflexion ?
    (cond
      ((null step) (setf *IARF-ET* *INF*)
       (early-time (rem-from shapes step))
       (MODEL2-A (rem-from shapes nil *INF*)))
      (t (setf *IARF-ET* step)
       (early-time (rem-from shapes step))
       (late-time (rem-from shapes nil step))))
  (get-early)
  (get-trans)
  (get-late))

;; MODEL4

(defun MODEL4 (shapes)
  (format stream "-3-%
  Early time is wellbore storage and skin.")
  (setf *MAX-WBS* *MAX1*)
  (setf *IARF-ET* (car (last (*segments*))))
  (format t "The test was stopped before the end of wellbore ~
  storage. A value for radial flow had to be assumed. The last data~
  was chosen")
  (get-early))

;; MODELS

(defun MODELS (shapes)
  (let ((step (car (member-shape 'STEP shapes))))
    (cond
      ((not (null step)) (early-time (rem-from shapes step))
       (setf *IARF-ET* step))
      (t (setf *IARF-ET* (car (last *segments*)))
       (format stream "The test was stopped before radial flow.~
       A value had to be assumed: the last data was chosen")))
  (get-early))

(defun get-early (shapes)
  (if (null *MAX-WBS*) (early-time shapes)
    (cond ((not (null *MAX-WBS*))
      (get-cde2s)
      (format stream "~S-~S
      CDe2s =~S
time match =~S
      pressure match =~S"
      *CDe2S* *tmatch* *pmatch*)
    (t (format stream "~S-~S Early time is fracture")))))

(defun get-trans ()
  (cond
    ((null *MIN-DPP*) nil)
    ;;!! The table used in this case is in natural log
    (t (let* ((der-min (+ (* (height *MIN-DPP*) (log 10))
      (log *pmatch*)))
      (tmin (* (expt 10 (caadr *MIN-DPP*)) *tmatch*))
      (omega (exp (lookup der-min *dermin-omega-table*)))
      (lambda (*(/ 1 (* tmin *cde2s*)) (log (/ 1 omega))))))
    (format stream "~S-~S
    Omega =~S
    Lambda= =~S")
  )

Omega = ~S
Lambda= ~S"
\[ \frac{\omega}{\Lambda} \]
(defun extraction (filename)
  "Extracts a list of shapes out of the digitized data in filename"
  (let ((list-of-points (pair filename)))
    (setf *inter* nil)
    (setf *segments* (extend (segmentation list-of-points)))
    (representation *segments*))

(defun model (shapes)
  "Determines the model for a given list of shapes"
  (interpretor n0 shapes))