RISK ANALYSIS AND DECISION-MAKING SOFTWARE PACKAGE
(1997 VERSION) USERS MANUAL

Final Report
September 1997

RECEIVED
MAR 11 1999
OSTI

By
Frank T-H Chung

February 1999

Performed Under Contract No. DE-AC22-94PC91008
(Original Report Number NIPER/BDM-0227 Rev. 1)

BDM Petroleum Technologies
BDM-Oklahoma, Inc.
Bartlesville, Oklahoma

National Petroleum Technology Office
U. S. DEPARTMENT OF ENERGY
Tulsa, Oklahoma

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government.

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DOE/PC/91008-0227
Distribution Category UC-122

Risk Analysis and Decision-Making Software Package (1997 Version)
Users Manual

By
Frank T-H Chung

February 1999

Work Performed Under Contract DE-AC22-94PC91008
(*Original Report Number NIPER/BDM-0227 Rev. 1*)

Prepared for
U.S. Department of Energy
Assistant Secretary for Fossil Energy

Rhonda P. Lindsey, Technology Manager
National Petroleum Technology Office
P.O. Box 3628
Tulsa, OK 74101

Prepared by:
BDM Petroleum Technologies
BDM-Oklahoma, Inc.
P.O. Box 2565
Bartlesville, OK 73005

TABLE OF CONTENTS

1.0	INTRODUCTION.....	1
2.0	OVERVIEW OF RISK ANALYSIS.....	3
2.1	What Is Risk?.....	3
2.2	Misconceptions about Risk Analysis.....	3
2.3	Advantages of Risk Analysis	4
2.4	Quantifying Risk	4
2.5	Risk Analysis Methods	5
3.0	GETTING STARTED	7
3.1	System Requirements.....	7
3.2	How to Install the Program	7
3.3	How to Run the Program.....	7
4.0	INVESTMENT RISK	11
4.1	What Is Gambler's Ruin?	11
4.2	Example of an Investment Risk (Gambler's Ruin).....	12
5.0	MONTE CARLO SIMULATION.....	15
5.1	What Is a Monte Carlo Simulation?.....	15
5.2	How to Run a Monte Carlo Simulation.....	16
5.3	Examples of Monte Carlo Simulations.....	20
6.0	BEST FIT FOR DISTRIBUTION FUNCTIONS	25
6.1	What Are Best Fit for Distribution Functions Used for?	25
6.2	How to Run Best Fit for Distribution Functions	26
6.3	Examples of Best Fit for Distribution Functions	27
7.0	SAMPLE OR RANK LINEAR CORRELATIONS.....	31
7.1	How Are Sample or Rank Linear Correlations Used?	31
7.2	How to Run Sample or Rank Linear Correlations.....	31
7.3	Example of Sample or Rank Linear Correlations.....	32
8.0	ENHANCED OIL RECOVERY (EOR).....	35
8.1	What Is EOR Methods Screening?	35

8.2	How to Run EOR Screening	35
8.3	Example of EOR Method Screening.....	36
9.0	ARTIFICIAL NEURAL NETWORKS	37
9.1	What Are Artificial Neural Networks?	37
9.2	How to Run an Artificial Neural Network.....	37
9.3	Example of an ANN.....	39
10.0	DECISION-TREE ANALYSIS.....	43
11.0	REFERENCES	45

APPENDICES

A	TERMINOLOGY.....	47
B	PROBABILITY DISTRIBUTION FUNCTIONS.....	49
C	ARTIFICIAL NEURAL NETWORKS	59

TABLES

4-1	Input Data for Investment Risk Analysis.....	12
6-1	Files for Best-Fit Program Testing.....	28
9-1	Log Values Used for Training.....	39
9-2	Determination of Lithology from Well Logs Using the Trained BPN	41

FIGURES

2-1	How Prediction Tools Are Implemented	6
3-1	Opening Screen Display	8
3-2	Analysis Tool Selection Screen.....	9
4-1	An Example of Investment Analysis.....	13
4-2	Result after Re-adjusting the Risk Money Value	13
5-1	LHC Sampling Method.....	16
5-2	Sampling Results by LHC (*) and Random (o) Sampling Methods.....	16
5-3	Interface for Monte Carlo Simulation Setting.....	17
5-4	Simulation Results (Expected Value) by (a) Random (MC) Sampling Method and (b) LHC Method with Various Iteration Numbers.....	18
5-5	Monte Carlo Simulation Input Interface.....	21
5-6	Monte Carlo Simulation Result	22
5-7	Monte Carlo Simulation Input for Environmental Risk Analysis	23
5-8	Environmental Risk Analysis Result.....	24
6-1	Opening Interface for Best-Fit Tool.....	25
6-2	Input of Porosity Data for Statistical Analysis.....	27
6-3	Result of Statistical Analysis.....	28
6-4	Input of a Probability Density Distribution (Gamma) Data.....	29
6-5	Best-Fit Result.....	30
7-1	Opening Interface for Correlation Setting	32
7-2	Correlation Plot for Porosity-Permeability Data.....	33
7-3	Porosity-Permeability Correlations	33
8-1	Interface for EOR Method Screening	35
8-2	Example of EOR Method Screening Results	36
9-1	Opening Interface for BPN Setting.....	37
9-2	Example of Data Input and Setting for BPN.....	40
9-3	Result of Training.....	41
C-1	Artificial Neural Network Node.....	59
C-2	Multilayer Feedforward Network	60
C-3	Multilayer Recurrent Network.....	60

1.0 INTRODUCTION

This manual provides instructions for using the U.S. Department of Energy's (DOE) risk analysis and decision making software (1997 version) developed at BDM Petroleum Technologies by BDM-Oklahoma, Inc. for DOE, under contract No. DE-AC22-94PC91008. This software provides petroleum producers with a simple, handy tool for exploration and production risk analysis and decision-making. It collects useful risk analysis tools in one package so that users do not have to use several programs separately. The software is simple to use, but still provides many functions. The 1997 version of the software package includes the following tools:

- ◆ Investment risk (Gambler's ruin) analysis
- ◆ Monte Carlo simulation
- ◆ Best fit for distribution functions
- ◆ Sample and rank correlation
- ◆ Enhanced oil recovery method screening
- ◆ Artificial neural network

This software package is subject to change. Suggestions and comments from users are welcome and will be considered for future modifications and enhancements of the software. Please check the opening screen of the software for the current contact information. In the future, more tools will be added to this software package.

This manual includes instructions on how to use the software but does not attempt to fully explain the theory and algorithms used to create it.

2.0 OVERVIEW OF RISK ANALYSIS

2.1 What Is Risk?

Risk contains two essential components: uncertainty and loss. If the outcome of an action is uncertain or uncontrollable and may cause some loss (e.g., of money, human life), the action is risky. The degree of risk is based on both the probability of failure and the outcome for each failure. For instance, buying a lottery ticket is a no-risk action because the loss is insignificant, even though the probability of winning is very low (usually less than one in 10 million). On the other hand, petroleum exploration is a high risk business because the loss is large for one failure in drilling, even though the success rate may be more than 10%. This distinction is important. If you judge the situation to be risky, risk becomes one criterion for decision-making—and risk analysis becomes viable.

Different businesses face different types of risk. The major types of risks faced by the petroleum industry are financial risks, safety risks, and environment risks.

- Financial risks focus on short- and long-term property and revenue loss, and investment returns.
- Safety risks focus on human safety and accident prevention, mainly within workplace boundaries. Safety risks are hardly estimated quantitatively.
- Environment risks concern the effects of human health, as well as ecosystem impacts, and the focus is on liability and insurance.

2.2 Misconceptions about Risk Analysis

Risk analysis (or risk assessment) is a process of estimating the probability of occurrence of an event and the magnitude of adverse effects—financial, safety, health, and ecological—over a specified time period. It aids decision making and can minimize losses.

One misconception about risk analysis is that risk analysis will eliminate risk in decision making. Indeed, risk and uncertainty cannot be eliminated from an event through any analysis method. Risk analysis tools do not reduce or eliminate risk; instead, they evaluate, quantify, and help you to understand risk so that you can design a decision strategy to minimize your exposure to risk.¹

Another misconception is that risk analysis methods can replace professional judgment. Risk analysis methods are intended to supplement, rather than replace, the necessary judgments. Personal experience and vision remain very important in decision making.

2.3 Advantages of Risk Analysis

Personal decision making is individual, inconsistent, vague, insubstantial, subjective, and can be influenced by personal emotion and other factors, such as political concerns. Also, the skill is usually not transferable. Risk analysis methods, however, can provide clearer, more substantial information for decision making. They can deal with very complicated problems that the human brain cannot handle. Therefore, decision making based on risk analysis tends to be more objective, consistent, and reliable. Risk analysis can reduce personal error and promote teamwork. It can also help companies win support from investors. More important, risk analysis can help prevent catastrophic loss due to human errors in judgment.

2.4 Quantifying Risk

Risk analysis is designed to deal with vague, fuzzy, and uncertain systems. The first step is to quantify the variables using one of two methods: the probability approach and the fuzzy logic approach. The uncertainty of variables can be expressed as probabilities or as fuzzy numbers.

Probability is based on the statistics of a large number of experimental results. For instance, a success rate for exploration can be based on exploration and drilling results in the same area or in other geologically similar areas. These uncertain variables can either be expressed as a deterministic value (e.g., 0.20) or as a distribution function (e.g., normal distribution).

A fuzzy number is represented by a set of elements and its associated membership function (denominator).^{2,3} For instance, the exploration success rate can be represented as

$$\text{success rate} = \{1.00/0.0, 0.30/0.1, 0.25/0.5, 0.20/0.8, 0.10/0.5, 0.00/0.1\}$$

This means that the possibility of a 100% success rate (1.00) is zero, of a 30% success rate (0.30) is 10%, and of a 25% success rate (0.25) is 50%. A 20% success rate is the most frequent outcome.

In the petroleum business, risks are usually judged by profitability. For instance, an enhanced oil recovery (EOR) project may not make a profit although it can recover more oil than before. Profitability computations usually include only direct investments, gross revenues, taxes, royalties, and operating costs. Many methods are used to measure profitability. (Appendix A defines some of the terms used to express profitability.) For more detailed explanations and calculations of profitability, see the discussion in reference 1 (see Section 11.0 for the list of references).

2.5 Risk Analysis Methods

In general, there are three different approaches that have been used in the risk-based decision process: (1) decision tree analysis, (2) stochastic simulation, and (3) artificial intelligence (AI) analysis methods. The applications of these three methods are based on the complexity of the problem.

Decision tree analysis is used for sequence decision making processes.¹ A diagram that looks like a tree branch has to be constructed to show all the subsequent possible events and decision options that are outcomes from previous decisions. This analysis method is used only for simple cases in which the anticipated events and the probability for each event are already known. Computations involved in this analysis are relatively simple and can be handled with calculators.

In many cases, the anticipated outcomes depend on several input variables whose values may not be known exactly. This kind of problem is usually analyzed by a **stochastic simulation** method, such as a Monte Carlo simulation.^{1,4,5} The inputs are probability distributions, and the output of such stochastic methods is also given in terms of distributions. In contrast, the output of decision-tree analysis gives single values.

As the complexity of a system increases, the conventional quantitative techniques of system analysis become more and more unsuitable. For instance, predicting crude oil prices or the change in the price of a stock requires the consideration of too many uncertain factors. These influential factors are very hard to model with formal mathematical tools. To tackle these problems that are hard to deal with by formal logical means, one must employ unconventional methods, such as artificial neural networks and fuzzy expert systems.

Artificial intelligence technologies have the capability of reasoning from fuzzy, noisy, and incomplete information. Recently, these unconventional technologies have been applied in areas such as geologic play appraisal, drilling problem diagnosis, production forecasting, reservoir characterization, and Wall Street stock prediction. This new version of the DOE Risk Analysis software package has included an artificial intelligence (AI) technology—artificial neural networks (ANN). The implementations of AI technology and other tools for prediction and forecast can be confusing for nonprofessional users. Figure 2–1 provides a flow diagram to illustrate the applications of various prediction tools.

Figure 2-1 How Prediction Tools Are Implemented

3.0 GETTING STARTED

3.1 System Requirements

Check that your computer meets these minimum system requirements:

- ◆ IBM-compatible 486 or Pentium PC
- ◆ Microsoft Windows 95, or higher, running in enhanced model (Windows 3.1.1 users will only be able to run Investment Risk Analysis and EOR Project Risk Analysis.)
- ◆ 8 MB RAM
- ◆ 8 MB available hard disk space
- ◆ VGA or SVGA color monitor

3.2 How to Install the Program

Follow this procedure to install the software:

1. Verify that there is at least 8 MB hard disk space on the drive where you plan to install the software and on the drive containing your Windows directory.
2. Insert Installation Disk 1 into your computer's diskette drive.
3. Go to the Windows File Manager and choose **RUN** from the **File** menu.
4. Type **A:SETUP** and click the **OK** button.
If your diskette is not in drive A, replace A with the appropriate drive letter.
5. Follow the prompts on the screen for installation.

You can change the software name or use the default name of "DOE-Risk."

If you encounter problems while installing the Risk Analysis software, please let us know. After completing the installation, you are ready to run the program.

3.3 How to Run the Program

Follow these steps to run the program:

1. Double click on the DOE-Risk icon.

NOTE: The opening interface (Fig. 3-1) will display on the screen. After about 10 seconds, a message box will pop up with a legal notice. If you have any questions about the legal notice, please contact us.

Figure 3-1 Opening Screen Display

2. Click on the **OK** button if you understand the legal notice.
The screen will change to the information interface which gives the information for contacting technical support.
3. Click on the **OK** button to change the screen to the tool selection interface (Fig. 3-2).
4. Select one of the following tools:

◆ Investment risk (Gambler's ruin) analysis	(See Section 4.0 for details on use.)
◆ Monte Carlo simulation	(See Section 5.0 for details on use.)
◆ Best fit for distribution functions	(See Section 6.0 for details on use.)
◆ Sample or rank correlation	(See Section 7.0 for details on use.)
◆ EOR method screening	(See Section 8.0 for details on use.)
◆ Artificial neural networks	(See Section 9.0 for details on use.)
◆ Decision-Tree Analysis	(Currently unavailable)
5. Click on the square (numbered) selection button to choose the tool.
6. Turn to the appropriate section (4.0, 5.0, 6.0, 7.0, 8.0, or 9.0) for details on how to use the software tool.

Figure 3-2 Analysis Tool Selection Screen

4.0 INVESTMENT RISK (GAMBLER'S RUIN) ANALYSIS

4.1 What Is Gambler's Ruin?

An important concern of investors, especially in a high-risk industry such as petroleum exploration, is the possibility of going broke through continuous failure. This risk is similar to gambling and is called "gambler's ruin."⁶ The first tool provided in this software package analyzes the possibility of gambler's ruin and estimates the minimum acceptable probability of success and the capital required to avoid the risk of gambler's ruin.

Investment risk analysis involves these basic parameters:

- ◆ Risk money (X) = the money loss for each failure (e.g. dry-hole drilling)
- ◆ Potential value of the reward (R) = the commercial value of each success
- ◆ Amount of available capital (C) = total investment or available funding for the adventure
- ◆ Probability of success of the venture (P) = estimated success rate
- ◆ Acceptable risk level (α_m) = the acceptable level of risk (i.e., the probability of going broke through continuous failure)

Based on the gambler's ruin theory presented by Arps and Arps,⁶ the minimum acceptable probability (P_m) of success for the venture breaking even in the long run is:

$$P_m = \left[1 - \frac{1n(1 + R / C - X / C)}{1n(1 - X / C)} \right]^{-1} \quad (4-1)$$

and the chance (α) of an investor going broke through a continuous string of failures is:

$$\alpha = (1 - P) C / X \quad (4-2)$$

The investment analysis tool will tell the user the

- ◆ Minimum acceptable probability of success
- ◆ Chance of going broke
- ◆ Minimum acceptable probability of success in order to meet the acceptable risk level

To use the tool, follow these steps:

1. Click on the first button on the Analysis Tool Selection screen (main menu).
2. Type the data requested into the first empty field on the "Gambler's Ruin" Analysis screen and press **Enter**.

3. Repeat Step 2 until all the required information has been entered.
4. Click on the **Run Analysis** button.

After finishing the analysis, click on **Exit** on the **File** menu to return to the **Analysis Tool Selection** screen.

4.2 Example of an Investment Risk (Gambler's Ruin)

An independent oil producer is planning to invest \$2 million in oil exploration and drilling (E&D) in an exploratory field. The average success rate of exploration and drilling in this field is 20%; the commercial value of each producing well is \$1 million. This means that the producer can sell the well right after it is completed. The estimated total cost for drilling a well, including exploration costs, is \$200,000. The company is willing to accept a risk level at or below 10% (the probability of going broke has to be less than 10%). The input data for the investment risk are listed in Table 4-1.

Table 4-1 Input Data for Investment Risk Analysis

Variables	Values
Risk money	\$200,000
Potential value of reward for each success	\$1,000,000
Available capital	\$2,000,000
Estimated probability of success	20%
Acceptable risk level (probability of gambler's ruin)	10%

After inputting these data and running the analysis, the results of the analysis (Fig. 4-1 on the next page) will promptly appear. The calculated minimum required probability of success for the investment to break even in the long run is about 23.84%, which is higher than the average success rate (20%), and the chance for the investor to broke is about 10.74%, which is also higher than the acceptable level (10%).

What can be done to bring the risk level within the acceptable level? Equation 4-2 shows that increasing the capital funding (C), or increasing the success rate (P), or reducing cost (X) can lower the chance of gambler's ruin. The program can solve this "what if" problem by trying different values of capital or the success rate. For example, by increasing the capital slightly (e.g., \$2,100,000), the risk level will drop below the maximum acceptable level (10%).

However, the minimum acceptable success rate will not be changed by raising capital. It is determined by the risk money and the potential value of each success. For example, by reducing the risk money from \$200,000 to \$170,000, users can change the result (see Fig. 4-2). In other words, if the producer can reduce the exploration and drilling costs to less than \$170,000 per well, the producer can meet the investment criteria.

"Gambler's Ruin" Analysis

File

A. Given Conditions :

	Value :
1. Risk money (Net loss for each failure), \$	200000
2. Potential value of reward for each success, \$	1000000
3. Available capital, \$	2000000
4. Estimated probability of success, %	20
5. Acceptable probability of Gambler's Ruin, % (= Acceptable risk level)	10

Run Analysis

B. Analysis Results :

1. Min. acceptable probability of success, % (To break even in the long run)	23.84624
2. Chance of investor's going broke (Gambler's Ruin), % (with the above conditions 1-4)	10.73742
3. Min. acceptable probability of success, % (with the acceptable probability of ruin- item 5)	20.56718

Figure 4-1 An Example of Investment Analysis

"Gambler's Ruin" Analysis

File

A. Given Conditions :

	Value :
1. Risk money (Net loss for each failure), \$	170000
2. Potential value of reward for each success, \$	1000000
3. Available capital, \$	2000000
4. Estimated probability of success, %	20
5. Acceptable probability of Gambler's Ruin, % (= Acceptable risk level)	10

Run Analysis

B. Analysis Results :

1. Min. acceptable probability of success, % (To break even in the long run)	20.37597
2. Chance of investor's going broke (Gambler's Ruin), % (with the above conditions 1-4)	7.242394
3. Min. acceptable probability of success, % (with the acceptable probability of ruin- item 5)	17.77573

Figure 4-2 Result after Re-adjusting the Risk Money Value

5.0 MONTE CARLO SIMULATION

5.1 What Is a Monte Carlo Simulation?

A Monte Carlo simulation^{4,5} is used to solve certain stochastic (involving a random variable) problems where the passage of time plays no substantive role. It is widely used to solve certain statistical problems that are not analytically tractable. The simulation requires that a mathematical relationship (model) be established between the event outcome (expectation) and influence factors (input variables). For example, to estimate the expectation of an outcome function $g(X)$, where $X = (x_1, x_2, \dots, x_n)$ denotes input variables, the simulation is used to estimate the expectation $E[g(X)]$. First, a sequence of N random vectors $X^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$ where $k=1, \dots, N$, are generated according to the probability distribution function, then the random variable $Y^{(k)} = g(X^{(k)})$ for each k is computed. Using N sample points, the expectation is approximated by

$$E[g(X)] = E(Y) = \lim_{N \rightarrow \infty} \frac{(Y^{(1)} + \dots + Y^{(N)})}{N} \quad (5-1)$$

Limitations. Monte Carlo is a random sampling method and requires a large sampling number N for accurate estimates. As a result, the method converges slowly. Its long computing time prohibits its application in reservoir simulation. Also, the large number of random samples of the input distribution used in the method tends to give a higher variance for the estimates and hence increases the uncertainty for the expected outcomes.

Managing Limitations. Several variance reduction techniques have been used to reduce the number of samples and better represent the variance.^{4,5} This software offers both Latin hypercube (LHC) and random (MC) sampling methods. The LHC sampling method reduces the number of samples and variance. In LHC (as shown in Fig. 5-1 on the next page), the range of each variable is divided into nonoverlapping intervals (m) on the basis of equal width or equal probability. These intervals are sampled according to probability density functions associated with the variables. Rather than sampling all possible combinations, the method selects only m of these combinations. Each stratum of each variable is only sampled once without replacement. Thus, the full range of each input variable is sampled. This can significantly improve the accuracy and convergence rate,⁷ as shown in the next example.

LHC Example. Suppose one desires to select five input vectors from two variables $X1$ and $X2$. In Figure 5-2 on the next page, the ranges of both variables are divided into five subregions. The asterisk (*) denotes a possible set of pairs $(X1, X2_i)$, $i=1,2, \dots, 5$, selected by LHC. It is clear that LHC has forced each of the five intervals to be represented once, and the entire range of both $X1$ and $X2$ has been covered. In contrast, random sampling may result in the selection of pairs as indicated by the open circle (o). The ranges of both $X1$ and $X2$ are not fully

covered. FORTRAN code and a detailed discussion of the LHC method was given by Iman et al.^{8,9}

Figure 5-1 LHC Sampling Method

Figure 5-2 Sampling Results by LHC (*) and Random (o) Sampling Methods

5.2 How to Run a Monte Carlo Simulation

To run a Monte Carlo simulation, follow these steps:

BEFORE YOU BEGIN: If you do not already have the distribution function for the variable, first see Section 6.0 (Best Fit for Distribution Functions) to generate this information. If you do not have the rank correlation coefficients, see Section 7.0 (Sample or Rank Correlation).

1. Select the Monte Carlo simulation tool on the Analysis Tool Selection screen by clicking on the second button. An interface for simulation setting (Fig. 5-3) will appear.

Figure 5–3 Interface for Monte Carlo Simulation Setting

- On the Simulation Setting interface, select the sampling method (either random or Latin hypercube) by clicking on the option inside the **Method** box.

If You Have	Then Choose This
Only a few (< 4) variables, a simple formula, or plenty of memory	Random sampling
Many variables, a complex formula, or not enough memory (receive memory error messages)	Latin Hypercube sampling

- Move the cursor to the empty box by **Sampling number** (the default number is 1,000), type in the number desired (should be from 50 to 3,000), and press **Enter**.

The number of samples should not be too small (less than 50), because accuracy can suffer. On the other hand, too many samples (more than 3,000) requires more memory and more computing time without significant improvements in accuracy. Figure 5–4 compares the results of various sampling numbers for random (MC) and Latin hypercube (LHC) methods. Note that 2,000 samples should be accurate enough for both sampling methods.

(a)

(b)

Figure 5-4 Simulation Results (Expected Value) by (a) Random (MC) Sampling Method and (b) LHC Method with Various Iteration Numbers

4. Move the cursor to the empty box by **No. of repeat sampling**, type in the number desired (the default is 1), and press **Enter**.

This number tells the program how many times in a row to run the simulation. If you wish to run the simulation twice, entering the number 2. Sometimes you have to keep the **Sampling number** low (lets say 1,000 at Step 3) because of memory problems, but you really want to run a full 3,000 iterations. Therefore you specify 1,000 for the **Sampling number** and 3 for the **No. of repeat sampling**.

5. Move the cursor to the empty box by **Numbers of random variables**, type in the number desired, and press **Enter**. After you enter the number, a grid table will become visible with the number of rows specified.

EXAMPLE: In this formula for calculating oil in place, there are five variables:

$$Y = x_1 * x_2 * x_3 * (1-x_4) / x_5$$

Oil-in-place is determined by area (x_1), pay thickness (x_2), porosity (x_3), water saturation (x_4), and formation volume factor (x_5).

6. Move the cursor to the empty box by **Random number generator seed**, type in the number desired, and press **Enter**. This value **must be a negative integer**. The program uses this number to generate random numbers for its statistical simulation. You might wish to change the number if you plan to run the simulation more than once.
7. Select the name and distribution of values for each random variable. (Click on the arrow (column 4) to drop down a menu of options, use the sliding button to locate the desired function, and click on the appropriate function to select it.)

The value for the random variable can be a deterministic (crisp) value, or it can be represented by a distribution function with best fitted parameters (see Section 6.0). This program provides the most commonly used distribution functions: crisp, uniform, lognormal, triangular, normal, loguniform, exponential, and Weibull.

8. For the distribution function you chose, enter the characterization parameters in the corresponding cell of columns 5–7 as:

◆ Crisp (a)	a is a parameter
◆ Uniform (a, b)	a = minimum value, b = maximum value
◆ Lognormal (a, b)	a = mean, b = standard deviation
◆ Triangular (a, b, c)	a = minimum, b = most likely, c = maximum
◆ Normal (a, b)	a = mean, b = standard deviation
◆ Loguniform (a, b)	a = minimum value, b = maximum value
◆ Exponential (a)	a = mean value
◆ Weibull (a, b)	a and b are parameters

Appendix B gives the probability density distribution functions and their properties for all these functions. (For normal distribution, the default minimum value is 0; you can change this value by entering a new number in the **Min. Value** column.)

9. To get the statistical information of the sampling for each random variable, check the variable in the check box column (the second column, labeled "x") and then click on **statistic** on the menu bar. A statistical result that includes the distribution of the samples of the sampling will be displayed.
10. **LATIN HYPERCUBE SAMPLING USERS ONLY:** If you choose the LHC sampling method, you can impose rank correlations on some of the random variables. (If you

wish to generate the rank correlation coefficients, see Section 7.0 and run the program in another window.) If you wish to apply the data on the correlations, click on the **Yes** option button and type in the number of pairs that have rank correlations. A table will appear for inputting the rank correlation coefficient for each pair. The input correlation coefficients have to be self-consistent and positive definite. The program will check the positive definite criterion and give an error message if they are not.

11. Move the cursor to the boxes under **Model for the expectation Y**. Type in the mathematical relationship (model) between the expected variable (**Y**) and the random variables (**x_i**).

To make it easy to see the meaning of the Y variable, type in an abbreviation of the physical variable name in the box after Y. For example, use OIP for oil-in-place, PV for pore volume, or EMV for expected monetary value.

Y OIP

Next, type the mathematical model in the box after the equal sign (=). Note that numbers that are normally subscript in the formula are typed in full-size. For example, if the formula is $x_1 * x_2 * x_3 * (1-x_4) / x_5$, type in:

= x1 * x2 * x3 * (1-x4)/x5

In this example, the oil-in-place is determined by area (x_1), pay thickness (x_2), porosity (x_3), water saturation (x_4), and formation volume factor (x_5).

12. To run the simulation, click on **RunMC** on the menu bar.

The screen will blacken briefly, and the program will check all input information. An error message will appear if any error is found. Next, the program will run a simulation and display the statistical results. The results include mean, variance, skewness, kurtosis, probability density, and cumulative distribution of the expected outcome (**Y**).

13. Print out the results by clicking on **Print form** on the menu bar.

To return to the Simulation Setting interface, click on the **Go Back** option on the **File** menu to return to setting the interface and rerun the simulation. To quit the simulation and return to the main menu, click on **Exit** on the **File** menu.

5.3 Examples of Monte Carlo Simulations

Example 1: Expected Oil Field Monetary Value

The first example uses a Monte Carlo simulation to evaluate the expected monetary value (EMV) of an oil field. The EMV is calculated by:

$$EMV = 7758 * x_1 * x_2 * x_3 * x_4 * x_5 * (1 - x_6) / x_7$$

where:

x_1 (oil price in dollars)	Triangular distribution with parameters (15.0, 17.0, 20.0)
x_2 (recovery factor)	Triangular distribution with parameters (0.3, 0.4, 0.5)
x_3 (area in acres)	Lognormal distribution with parameters (200.0, 50.0, 0.0,)
x_4 (pay thickness in feet)	Lognormal distribution with parameters (15.0, 5.0, 0.0)
x_5 (porosity)	Normal distribution with parameters (0.35, 0.25, 0.00)
x_6 (water saturation)	Uniform distribution with parameters (0.3, 0.4)
x_7 (FVF)	Crisp distribution with parameters (1.14)

Assume there is rank correlation between variables x_5 and x_6 with a rank correlation coefficient = 0.5.

Figure 5-5 shows the input interface, and Figure 5-6 shows the simulation result.

Monte Carlo simulation

File Statistic RunMC

Simulation setting

Method:

☐ Random sampling

☒ Latin Hypercube sampling

Sampling Number : 1000 ($\geq 50; \leq 3,000$)

No. of repeat sampling : 1

Numbers of random variables: 7 (< 9)

Random number generator seed : -1 (Negative integer)

Input of random variables:

Xi	Name	Distribution	Para (a)	Para (b)	Para (c)	Min. Value	Max. Value
3	Area	6.Lognorm(a,b)	200.0000	50.0000	0.0000	0.0000	0.0000
4	Pay	6.Lognorm(a,b)	15.0000	5.0000	0.0000	0.0000	0.0000
5	Poro	5.Normal(a,b)	0.3500	0.2500	0.0000	0.0000	0.0000
6	Sw	2.Uniform(a,b)	0.3000	0.4000	0.0000	0.0000	0.0000
7	FVF	1.Crisp [a]	1.1400	0.0000	0.0000	0.0000	0.0000

Rank Correlation of random variables ? ☐ No ☒ Yes No. of correlated pairs: 1

Model for the expectation Y:

Y EMV = 7758 * x1 * x2 * x3 * x4 * x5 * (1 - x6) / x7

x(i)	x(j)	Corr.
5	6	0.50

Figure 5-5 Monte Carlo Simulation Input Interface

Figure 5-6 Monte Carlo Simulation Result

Example 2: Evaluating Environmental Risk

The second example uses a Monte Carlo simulation to evaluate environmental risk. In this simple example, the carcinogenic risk from the incidental ingestion of water from soil containing benzene is given by the formula:

$$Risk = K \times C \times SF$$

where K = factor including variables such as frequency of exposure, body weight, etc. (day^{-1})

C = concentration in soil (mg/kg)

SF = carcinogen potency factor (also called slope factor) (kg.day/mg)

The three variables have to be analyzed statistically. Assuming that the analysis results of the three variables are given as

K = normal distribution (4.5×10^{-5} , 1.2, 0.0)

C = lognormal distribution (1.25, 0.40, 0.0)

SF = lognormal distribution (0.029, 0.005, 0.0)

The input for the Monte Carlo simulation appears in Figure 5-7. (Note that the K-values have been scaled by multiplying 10^5 . Therefore, one must divide the result by 10^5 .) The calculated result of the probability distribution of carcinogenic risk appears in Figure 5-8. EPA has defined acceptable risks for carcinogens as within the range of 10^{-4} to 10^{-6} excess lifetime cancer risk. Figure 5-8 shows that the mean value of the risk is 1.632×10^{-6} and the maximum value is 0.6156×10^{-5} , which fall within the EPA's defined acceptable range.

Monte Carlo simulation

File Statistic RunMC

Simulation setting

Method:

☒ Random sampling

☐ Latin Hypercube sampling

Sampling Number : 1000 ($\geq 50; \leq 3,000$)

No. of repeat sampling : 1

Numbers of random variables: 3 (< 9)

Random number generator seed : -1 (Negative integer)

Input of random variables:

Xi	Name	Distribution	Para [a]	Para [b]	Para [c]	Min. Value	Max. Value
1	K	5.Normal[a,b]	4.5000	1.2000	0.0000	0.0000	0.0000
2	C	6.Lognorm[a,b]	1.2500	0.4000	0.0000	0.0000	0.0000
3	SF	6.Lognorm[a,b]	0.0290	0.0050	0.0000	0.0000	0.0000

Model for the expectation Y:

Y Risk = $x_1 \cdot x_2 \cdot x_3 / 10^5$

Figure 5-7 Monte Carlo Simulation Input for Environmental Risk Analysis

Figure 5-8 Environmental Risk Analysis Result

6.0 BEST FIT FOR DISTRIBUTION FUNCTIONS

6.1 What Are Best Fit for Distribution Functions Used for?

This tool is designed to analyze sample data and to find the best fitting function and its parameters. The tool supports the Monte Carlo program by supplying input, such as the distribution function and parameters for the input random variables.

The tool contains two program options: **Sample data** and **Probability density data**. If you have raw data (physical measurements), first run the **sample data analysis** to obtain the probability density distribution (data points). **Sample data** performs statistical analysis on a set of measured or sampled data for a variable and finds the data points for you.

Once you have data points, you are ready for the **density distribution function fitting**. Enter data points (the density distribution) into the **Probability density data** program. Then fit the data with an appropriate function to find the best-fitted function (e.g., normal, lognormal, gamma, etc., functions) to generate the function's characterization parameters. Later, enter the distribution function into the Monte Carlo tool (Section 5.0).

Best-fit for distribution functions

File Edit Run

Data Type:

☐ Sample data

☐ Probability density data

Data Table:

No. of data points =

Select distribution functions:

Give initial guess for parameters:

a: b: c:

X	Frequency
---	-----------

Figure 6-1 Opening Interface for Best-Fit Tool

6.2 How to Run Best Fit for Distribution Functions

A. To do sample data analysis, follow these steps at the interface menu for Best Fit for Distribution Functions:

1. Click on **Sample data** option in the **Data Type** box. (A one-column grid table opens.)
2. Move the cursor to the box by **No. of data points**, type in the number of data points, and press **Enter**.
3. Type the data values into the grid table, and press the **Enter** key after each entry.
If the data already are saved in a file, click on **Load Data** on the **File** menu to load the data into the grid table. You can also add or delete data to/from the grid table by clicking on **Edit** from the menu bar.
4. Click on the **Run** button to obtain the probability density distribution (data points) and display the graph.
5. Print out the result by clicking on **Print form** in the **File** menu or record the results for future reference. (At this time there is no feature to save the data for later reloading.)
6. Turn to Appendix B to view the possible functions and to see which function most resembles the one on your screen. Record the name of the function for use in Procedure B (Density Distribution Function Fitting), Step 5.
To return to the interface screen, click on **File | Go Back**. To clear all data, click on **File|Clear Data**.

B. For density distribution function fitting, follow these steps at the interface menu for Best Fit for Distribution Functions:

1. Click on the **probability density data** option in the data type selection box.
2. If there is data left from a previous analysis, click on **File|Clear Data** to clear all previous data.
3. Enter the number of data points, and type the data into the grid table (two columns). Or, if the probability density data are already saved in a file, click on **File | Load Data** to load the data into the grid table.
4. Type in the minimum and maximum range for the plot.
NOTE: You must specify the smallest and largest x point in order to see a graph.
5. Select the distribution function to be fitted by clicking on the draw-down arrow from the select distribution function box.
6. Give initial guess for the distribution function parameters (a,b,c).
7. Click on **Run** on the menu bar to display results.

8. Edit and analyze the data again, or click on **File | Save** to save the data.
9. Record parameters a , b , and maybe c for Monte Carlo analysis or other applications.

The program includes the following distribution functions:

◆ Beta	(a , b)	◆ Exponential	(a)
◆ Gamma	(a , b)	◆ Geometric	(a , b)
◆ Lognormal	(a , b)	◆ Normal	(a , b)
◆ Triangular	(a , b , c)		

6.3 Examples of Best Fit for Distribution Functions

A. Sample Data Analysis

A collected porosity data for an oil field is stored in the attached data file: **porosity.dat**. To load this data, click on **File|Load Data** on the menu bar, select the file **porosity.dat** from file directory, and click on **OK**. Figure 6–2 shows the loaded sample data. To perform statistical analysis for the data, click on **Run** on the menu bar.

Best-fit for distribution functions

File Edit Run

Data Type:

☒ Sample data

☐ Probability density data

Data Table:

No. of data points = 100

	X
1	0.2369
2	0.1004
3	0.2523
4	0.3730
5	0.3118
6	0.2191
7	0.2456
8	0.3965
9	0.2148
10	0.3061
11	0.3173
12	0.1047

Figure 6–2 Input of Porosity Data for Statistical Analysis

The result will appear promptly (see Figure 6–3) to show that the probability density of the sample porosity data has a normal distribution. The output contains the statistical characteristic parameters, probability density distribution, and cumulative distribution.

Figure 6–3 Result of Statistical Analysis

B. Distribution Function Analysis

To demonstrate a fitting of distribution function, the program supplies five data files of commonly used distribution functions for testing (see Appendix B for more information on distributions). The files are described in Table 6–1.

Table 6–1 Files for Best-Fit Program Testing

File Name	Description
beta.dat	Probability density distribution data which was generated by the Beta distribution function with the parameters $a = 3.0$, $b = 2.0$.
exponent.dat	Exponential distribution data with the parameter $a = 1.0$.
gamma.dat	Gamma distribution data with the parameters $a = 4.5$, $b = 2.5$.
lognormal.dat	Lognormal distribution data with the parameters $a = 15.0$, $b = 5.0$.
normal.dat	Normal distribution data with parameters $a = 0.25$, $b = 0.8$.

You can load any of these data sets by clicking on **File|Load Data** from the menu bar, selecting the data file from the file directory, and then clicking on **OK**. Figure 6-4 shows the loaded sample data (**gamma.dat**). Following the procedure described in the previous section (6.2-B), you should obtain the best-fitted function (i.e., gamma function) and the two characterization parameters as shown in Figure 6-5. You can try other functions or give different initial values for parameters a and b to view other distribution results.

Best-fit for distribution functions

File Edit Run

Data Type:

☐ Sample data

☒ Probability density data

Select distribution functions:

4. Gamma [a, b]

Give initial guess for parameters:

a : 4.0 b : 2.0 c :

Data Table:

No. of data points = 10

Min. Max.

X range (for plot) = 1.0 28.0

	X	Frequency
1	1.0000	0.0010
2	4.0000	0.0360
3	7.0000	0.0770
4	10.0000	0.0810
5	13.0000	0.0610
6	16.0000	0.0380
7	19.0000	0.0210
8	22.0000	0.0100
9	25.0000	0.0050
10	28.0000	0.0020

Figure 6-4 Input of a Probability Density Distribution (Gamma) Data

Figure 6-5 Best-Fit Result

7.0 SAMPLE OR RANK LINEAR CORRELATIONS

7.1 How Are Sample or Rank Linear Correlations Used?

This tool is used to find correlation coefficients for a selected pair of sampling data. Two types of correlation coefficients can be calculated¹⁰:

1. Sample linear correlation coefficient (r), also called the Pearson product-moment correlation coefficient:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \quad (7-1)$$

where $\bar{x}$ is the mean of the x_i 's, $\bar{y}$ is the mean of the y_i 's.

2. Rank correlation coefficient (r_s), also called the Spearman rank correlation coefficient:

$$r_s = \frac{\sum_i (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2} \sqrt{\sum_i (S_i - \bar{S})^2}} \quad (7-2)$$

where R_i is the rank of x_i among the other x 's, S_i is the rank of y_i among the other y 's, bars are assigned the appropriate midrank.

7.2 How to Run Sample or Rank Linear Correlations

To run Monte Carlo simulation analysis, you need to take into account the rank correlations among those variables in the model. You can use this fourth tool to analyze and obtain rank correlation coefficient for two variables. To use this tool, click the fourth button in the Analysis Tool Selection screen. This screen will display an empty grid box (shown in Figure 7-1 on the next page).

1. Move the cursor to the box by **Number of data points**, type in the number of data points, and press **Enter**. A grid table will appear.

Figure 7-1 Opening Interface for Correlation Setting

2. Type the data sets into the grid table and press a cursor key to input the data and move between cells. Or, if the data are already saved in a file, click on **File|Load Data** to load the data into the grid table.
3. Type in the columns of data sets which are to be analyzed. For example, type **1** in the empty box for x-1, and type **2** for x-2.
4. Click on the check boxes for both **Rank-order correlation coefficient (Spearman)** and **Linear correlation coefficient (Pears)**.

7.3 Example of Sample or Rank Linear Correlations

This example finds the correlation for a set of porosity-permeability data (shown in Fig. 7-2). The data set is stored in the data file **porperc.dat**. Click on **File|Load Data** to load the data into the grid table. Select columns x-1 and x-2, and click on the check box for **Rank-order correlation coefficient (Spearman)** and **Linear correlation coefficient (Pears)**. The result (Fig. 7-3) shows that the coefficients are about 0.456 for rank correlation and 0.472 for linear coefficient.

Figure 7-2 Correlation Plot for Porosity-Permeability Data

Figure 7-3 Porosity-Permeability Correlations

8.0 ENHANCED OIL RECOVERY (EOR) METHODS SCREENING

8.1 What Is EOR Methods Screening?

This tool offers a preliminary screening for EOR methods and helps you select an appropriate EOR method.

8.2 How to Run EOR Screening

Click on the fifth button on the Analysis Tool Selection screen, and the interface screen (Fig. 8-1) will appear. Follow these steps:

EOR METHOD PRELIMINARY SCREENING

Data Input:
(If you don't have the information for the highlighted parameter, press ENTER to skip it)

<u>Fluid Information:</u>	<u>Reservoir Conditions:</u>
API oil gravity (degrees) : <input type="text"/>	Temperature (F) : <input type="text"/>
In-situ oil viscosity (cp) : <input type="text"/>	Pressure (psig) : <input type="text"/>
Total dissolved solids (ppm) : <input type="text"/>	Oil saturation (fraction) : <input type="text"/>

Reservoir Formation Information:

Depth (ft) : <input type="text"/>	Pay zone thickness (ft) : <input type="text"/>
Porosity (fraction) : <input type="text"/>	Permeability, average (md) : <input type="text"/>
Rock type (1- Sandstone, 2- Carbonate) : <input type="text"/>	

Figure 8-1 Interface for EOR Method Screening

1. Type in the appropriate information into each empty box, then press **Enter** to input the information and advance to the next empty box. Fill all the empty boxes.
2. Click on the **Execute** button at the bottom of the screen.

Information about an applicable EOR method will appear, as will comments on other methods. The screening criteria are based on an National Petroleum Council report¹⁴ and National Institute for Petroleum Energy Research (NIPER) experience.

8.3 Example of EOR Method Screening

A sandstone reservoir in the Mid-Continent nears its economic limit of oil production through waterflooding. Which tertiary oil recovery processes should one consider to continue oil production? Input the following reservoir conditions on the interface screen and see Figure 8-2. The results screen lists applicable processes and explains why others do not apply.

Rock Properties		Oil Properties	
Permeability (md)	100	API gravity (degree)	38.7
Porosity	0.20	Viscosity at 113 F, 1400 psi (cp)	2
Pay thickness (ft)	25	Miscibility pressure (psi)	1,300
Reservoir depth (ft)	4,000		
Reservoir temperature (°F)	113		
Reservoir pressure (psig)			
Initial	2,100		
Current	1,400		
Residual oil saturation	0.35		
Well spacing (acres)	40		
Salinity (ppm)	50,000

Figure 8-2 Example of EOR Method Screening Results

9.0 ARTIFICIAL NEURAL NETWORKS

9.1 What Are Artificial Neural Networks?

Artificial neural networks (ANNs) are systems that are constructed to use some organizational principles resembling those of the human brain. They are a new generation of information processing systems that demonstrate the ability to learn, recall, and generalize from training patterns or data. ANNs are good at tasks such as pattern matching and classification, data clustering, and forecasting. For more background on ANNs, see Appendix C.

There are more than 40 functioning ANN models. This software uses the backpropagation neural network (BPN), the most widely used feedforward neural network system. The term backpropagation refers to the training method by which the weights of the network connection are adjusted. The calculations procedure is feedforward, from input layer through hidden layers to output layer. During training, the calculated outputs are compared with the desired values, and then the errors are backpropagated to correct all weight factors.

9.2 How to Run an Artificial Neural Network

To run the BPN program, click the sixth button in the Analysis Tool Selection screen. Follow these steps at the ANN interface (Fig. 9-1) screen.

The screenshot shows a software window titled "Forecast - Neural Networks (Backpropagation)". Inside, there is a dialog box titled "Artificial Neural Network Setting". The dialog box has several sections:

- Mode:** Two radio buttons, "Training" (selected) and "Prediction".
- Network Structure:** Four input fields for "Numbers of neurons": "Input layer", "Hidden layer", "Output layer", and "No. of samples". Below these are "Threshold value" (set to 0.0) and "Transfer function #" (set to 1).
- Training Parameters:** Four input fields: "Learning rate" (0.9), "Momentum coef." (0.6), "Max. iteration" (1000), and "Convergence criterion" (0.001).
- Input Data:** A table with 4 columns labeled x1, x2, x3, and x4. Below the table is a section for "Input-Weight Factors" with radio buttons for "No" (selected) and "Yes", and a table with 4 columns labeled 1, 2, 3, and 4.

Figure 9-1 Opening Interface for BPN Setting

A. For Training

1. Click on the option button of **Training** in the **Mode** box.
2. Move the cursor to an empty box under **Network Structure**, type in the desired number of layers, and press **Enter**. Fill in the empty boxes are for the following parameters:
 - **Input layer**
 - **Hidden layer**
 - **Output layer**
3. Move the cursor to the empty box by **No. of samples**, type the number of training samples, and press **Enter**. The input data table will show the grids with the rows (number equal to the number of samples plus two) and columns (equal to the sum of input and output layers).
4. Move the cursor to the empty box by **Threshold value**, type in the threshold value (default = 0.0), and press **Enter**.
5. Move the cursor to the box by **Transfer function** (default = 1), type in the threshold function number (1 = Log-Sigmoid function, 2 = Hyperbolic Tangent Sigmoid function, 3 = Linear function, 4 = Hard Limit function, and Symmetrical Hard Limit function), and press **Enter**.
6. Specify the Training Parameters (accept the default values or change them).
7. Type the training data into the input table, plus the data range in the last two rows (minimum and maximum values). Use the cursor key to register the numbers in each cell and also to move between cells.

First insert data for input layers at the beginning of the column, followed by data for output layers afterward. Note that the range is for the minimum and maximum of possible physical values—not the smallest and largest values in the current data set.
8. For initial Weight Factors, select the default **NO** option (the program will generate random numbers as initial input). If user want to give initial values, select **Yes** option, and type the initial values.
9. After complete the setting and data input, click on the **Run** button. The program may take a while to execute the calculations before the result is displayed. The waiting time depends on the number of iterations specified by users in the Training Parameter box. The training result will give the optimized weight factors and the BPN predicted values of neural network output.
10. Print out the input weight factors to use as the input for Procedure B (Prediction).

B. For Prediction

After you have trained the neural network, you can use the optimized weight factors to predict. To do prediction, follow these steps:

1. Click on the **Prediction** option button in the **Mode** box.
2. Specify the network structure (use the same numbers as that used for Step 2 of the Training procedure).
3. Type in the number of data points to be predicted and their range (minimum and maximum values), and press **Enter** after completing each box.
4. Select **Yes** in the box by weight factor input and type the weight factors into the table obtained from Training results.
5. Click on **Run** on the menu bar. The program will execute quickly and display results.

9.3 Example of an ANN

Applications of artificial neural networks have become pervasive in many areas. One example given is the use of a BPN for determining lithology from well logs. This simple example is designed to show you of how to use the tool. (The example is not designed to demonstrate the applicability and accuracy of this tool.) At the beginning, the BPN was trained with the following few data samples.¹³

Table 9-1 Log Values Used for Training

Samples	Logs			Lithology			
	Gamma Ray	Neutron	Density	Limestone	Dolomite	Shale	Sandstone
1	5.0	10.0	10.0	1	0	0	0
2	5.0	20.0	2.0	0	1	0	0
3	145.0	38.0	15.0	0	0	1	0
4	5.0	15.0	23.0	0	0	0	1
5	15.0	-2.0	-2.0	1	0	0	0
6	22.0	18.0	0.0	0	1	0	0
7	135.0	24.0	4.0	0	0	1	0
8	30.0	6.0	10.0	0	0	0	1
min.	0	-10	-10				
max.	150	40	40				

$$X_N = \frac{x - x_{\min.}}{x_{\max.} - x_{\min.}} \quad (9-1)$$

$$X_N = \frac{x - x_{\min.}}{x_{\max.} - x_{\min.}} \quad (9-1)$$

Figure 9-2 shows the BPN setting and input data. Users can change each of the settings, such as maximum iteration number. Figure 9-3 shows the result after 10,000 iterations.

[illegible]

Figure 9-2 Example of Data Input and Setting for BPN

Figure 9-3 Result of Training

After training, the neural network was used to determine the lithologies from gamma-ray, neutron, and density logging information. Here, a few data points were used to test the training result. Table 9-2 shows the predicted result and comparison with the literature.

Table 9-2 Determination of Lithology from Well Logs Using the Trained BPN

Gamma Ray	Neutron	Density	Predicted Lithofacies (reported)
23.0	27.0	5.0	Dolomite (Dolomite)
120.0	24.0	10.0	Shale and limestone mix (Shale)
20.0	2.0	0.0	Limestone (Limestone)
35.0	18.0	25.0	Sandstone (Sandstone)

10.0 DECISION-TREE ANALYSIS

This program is under development.

11.0 REFERENCES

1. Newendorp, P. D. 1975. *Decision Analysis for Petroleum Exploration*. Tulsa, Oklahoma: PennWell Books.
2. Zadeh, L. A. 1975. The Concept of Linguistic Variable and Its Application to Approximate Reasoning. *Information Sciences* 8: 199–249, 301–357.
3. Zadeh, L. A., and J. Kacprzyk (editors). 1992. *Fuzzy Logic for the Management of Uncertainty*. New York: John Wiley & Sons.
4. Law, A. M., and W. D. Kelton. 1991. *Simulation Modeling & Analysis*, 2nd ed. New York: McGraw-Hill, Inc..
5. Rubinstein, R. Y. 1981. *Simulation and the Monte Carlo Method*. New York: John Wiley & Sons.
6. Arps, J. J, and J. L. Arps. 1974. Prudent Risk-Taking. *Journal of Petroleum Technology* 27(7): 711–716.
7. McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 21 (2): 239–245.
8. Iman, R. L., and M. J. Shortencarier. 1984. *A FORTRAN 77 Program and User's Guide for the Generation of Latin HyperCube and Random Samples for Use with Computer Models*. Sandia Laboratory Report SAND 83-2365.
9. Iman, R. L., J. M. Davenport, and D. K. Zeigler. 1980. *Latin Hypercubic Sampling (Program User's Guide)*. Sandia Laboratory Report SAND 79-1473.
10. Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1986. *Numerical Recipes*. Cambridge, England: Cambridge University Press.
11. Demuth, H. B., M. H. Beale, and M. T. Hagan. 1996. *Neural Network Design*. Florence, Kentucky: PWS Publication.
12. Baughman, D. B. and Y. A. Liu. 1995. *Neural Networks in Bioprocessing and Chemical Engineering*. San Diego, California: Academic Press.

13. Rogers, S. J., J. H. Fang, C. L. Karr, and D. A. Stanley. 1992. Determination of Lithology from Well Logs Using a Neural Network. *AAPG Bulletin* 76 (5): p. 731–739.
14. National Petroleum Council. 1984. *Enhanced Oil Recovery*. National Petroleum Council report.

APPENDIX A

TERMINOLOGY

Payout—the length of time required to receive accumulated net revenues equal to the investment. Payout time is an approximate measure of the rate at which cash flows are generated early in the project. It can be expressed in terms of before-tax or after-tax revenues. It does not consider the total profit from the investment project.

Profit-to-investment ratio—the ratio of total undiscounted net profit to investment. It is a dimensionless number relating the amount of new money generated from an investment project per dollar invested. It is sometimes called the **return-on-investment (ROI)**.

Rate of return—the interest rate which makes the present value of net receipts equal to the present value of the investments. The calculation is a trial-and-error process which begins by selecting an interest rate and discounting all the cash flows back to time zero. The rate of return concept introduces the “time-value” of money into the profitability criterion.

Net present value (NPV)—the net cash flow of previous years that has been converted to the present values by a discount factor (or interest rate). Net Present Value has all the features of rate of return. If $NPV = 0$, then the investment is yielding a rate of return equal to the discount factor used. If NPV is negative, it means that the rate of return is less than the discount factor used. If NPV is positive, it means that the investment will earn a rate of return equal to the discount factor plus an additional amount of cash. NPV is a commonly used profit criterion.

Discounted profit-to-investment ration (DPR)—the ratio of NPV to the present value of investment.

Expected monetary value (EMV)—the product of the probability of occurrence of the outcome and the resulting monetary profits or losses.

APPENDIX B

PROBABILITY DISTRIBUTION FUNCTIONS

Name:	BETA (a, b)
Density distribution:	$f(x) = \frac{x^{a-1}(1-x)^{b-1}}{\beta(a, b)}$
where	$\beta(a, b) = \int_0^1 t^{a-1}(1-t)^{b-1} dt$
Cumulative distribution:	(No closed form)
Parameters:	$a > 0, b > 0$
Domain:	$0 \leq x \leq 1$
Mean:	$\frac{a}{a+b}$
Mode:	$\frac{a-1}{a+b-2} \quad \text{if } a > 1, b > 1$ $0, 1 \quad \text{if } a < 1, b < 1$ $0 \quad \text{if } a < 1, b \geq 1 \text{ or } a = 1, b > 1$ $1 \quad \text{if } a \geq 1, b < 1 \text{ or } a > 1, b = 1$
Variance:	$\frac{ab}{(a+b)^2(a+b+1)}$
Function behavior:	(See next page)

Name: EXPON (a)

Density distribution: $f(x) = \frac{1}{a} \exp(-x/a)$

Cumulative distribution: $F(x) = 1 - \exp(-x/a)$

Parameters: $a > 0$

Domain: $x \geq 0$

Mean: a

Mode: 0

Variance: a^2

Function behavior:

Name: GAMMA (a, b)

Density distribution:
$$f(x) = \frac{b^{-a} x^{a-1} \exp(-\frac{x}{b})}{\Gamma(a)}$$

Cumulative distribution: (No closed form)

Parameters: $a > 0, b > 0$

Domain: $x \geq 0$

Mean: ab

Mode: $b(a-1)$ if $a \geq 1$
 0 if $a < 1$

Variance: ab^2

Function behavior::

Name: LOGNORM (a, b)

Density distribution:
$$f(x) = \frac{1}{x\sqrt{2\pi b^2}} \exp\left[-\frac{(\ln x - a)^2}{2b^2}\right]$$

Cumulative distribution: (No closed form)

Parameters: $a, b > 0$

Domain: $x > 0$

Mean: $\exp\left(a + \frac{b^2}{2}\right)$

Mode: $\exp(a - b^2)$

Variance: $\exp(2a - b^2)[\exp(b^2) - 1]$

Function behavior:

Name: NORMAL (a, b)

Density distribution: $f(x) = \frac{1}{\sqrt{2\pi b^2}} \exp\left[-\frac{(x-a)^2}{2b^2}\right]$

Cumulative distribution: (No closed form)

Parameters: $a, b > 0$

Domain: $-\infty < x < \infty$

Mean: a

Mode: a

Variance: b^2

Function behavior:

Name: TRIANGULAR (a, b, c)

Density distribution: $f(x) = \frac{2(x-a)}{(b-a)(c-a)}$ if $a \leq x \leq b$

$f(x) = \frac{2(c-x)}{(c-a)(c-b)}$ if $b < x \leq c$

Cumulative distribution: $F(x) = 0$ if $x < a$

$F(x) = \frac{(x-a)^2}{(b-a)(c-a)}$ if $a \leq x \leq b$

$F(x) = 1 - \frac{(c-x)^2}{(c-a)(c-b)}$ if $b < x \leq c$

$F(x) = 1$ if $c < x$

Parameters: $a = \min$, $b = \text{most likely}$, $c = \max$.

Domain: $a \leq x \leq c$

Mean: $\frac{a+b+c}{3}$

Mode: b

Variance: $\frac{a^2 + b^2 + c^2 - ab - ac - bc}{18}$

Function behavior:

Name: UNIFORM (a, b)

Density distribution: $f(x) = \frac{1}{b-a}$

Cumulative distribution: $F(x) = \frac{x-a}{b-a}$

Parameters: $a \leq b$

Domain: $a \leq x \leq b$

Mean: $\frac{a+b}{2}$

Mode: (No)

Variance: $\frac{(b-a)^2}{12}$

Function behavior:

Name:

WEIBULL (a, b)

Density distribution:

$$f(x) = ab^{-a} x^{a-1} \exp[-(\frac{x}{b})^a]$$

Cumulative distribution:

$$F(x) = 1 - \exp[-(\frac{x}{b})^a]$$

Parameters:

$$a > 0, \quad b > 0$$

Domain:

$$x > 0$$

Mean:

$$\frac{b}{a} \Gamma(\frac{1}{a})$$

Mode:

$$b[\frac{a-1}{a}]^{\frac{1}{a}} \quad \text{if } a \geq 1$$

$$0 \quad \text{if } a < 1$$

Variance:

$$\frac{b^2}{a} [2\Gamma(\frac{2}{a}) - \frac{1}{a} \Gamma(\frac{1}{a})^2]$$

Function behavior:

APPENDIX C

ARTIFICIAL NEURAL NETWORKS

Two primary elements make up a neural network: processing elements (called nodes or units) and interconnections. The network mimics the human brain, which contains more than 10 billion (biological) neurons. Hence, the processing elements in ANNs are also called artificial neurons. An ANN node model of a biological neuron is shown in Figure C-1.

Figure C-1 Artificial Neural Network Node

In this model, the j -th processing element computes a weighted sum of its inputs and outputs y_j according to whether this weighted input sum is above or below a certain threshold T_j :

$$y_j = f\left(\sum_i^n (w_{ij}x_i) - T_j\right) \quad (C-1)$$

where the function f is called transfer function. The most commonly used transfer function is the sigmoid (S-shape) function. A typical sigmoid function is:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (C-2)$$

Other types of functions such as hard limit, symmetrical hard limit, linear, and hyperbolic tangent are commonly used.¹¹

C1.0 Neural Network Structures

The structure of the neural network is defined by the interconnection architecture between the processing elements. The basic types of structures are feedforward and recurrent nets (see Figs. C-2 and C-3). Others are combinations of these two types.¹¹

Figure C-2 Multilayer Feedforward Network

Figure C-3 Multilayer Recurrent Network

C2.0 LEARNING RULES

The primary training method commonly used is Error-Correction Learning. It is a form of supervised learning where the weights are adjusted in proportion to the output-error vector, ϵ . The output error from the k -th node on the output layer is defined as:

$$\epsilon_k = d_k - c_k \quad (\text{C-3})$$

where d_k is the desired output, and c_k is the calculated output, for the k -th node on the output layer only. The total squared error on the output layer, E , is:

$$E = \sum_k \varepsilon_k^2 = \sum_k (d_k - c_k)^2 \quad (\text{C-4})$$

Knowing E , we can calculate the change in the weight factor for the i -th connection to the j -th node, W_{ij} :

$$w_{ij\text{new}} - w_{ij\text{old}} = \Delta w_{ij} = \eta_j a_i E \quad (\text{C-5})$$

where η_j is a linear proportionality constant for node j , called the learning rate (typically, $0 < \eta_j < 1$), and a_i is the i -th input to node j .

C3.0 Neural Network Design

With more than 40 functioning models to choose from, it is important to know which models have had the most success and to understand their similarities and differences. After choosing the model, you then have to decide the number of hidden layers and the nodes for each layer. The sizes (number of nodes) of input and output layers are fixed by the number of inputs and outputs used. The sizes of middle (hidden) layers are determined by trial and error. It is better to choose the smallest number of neurons possible for a given problem to allow for generalization. If there are too many neurons, the net will tend to memorize patterns. The number of neurons may be dictated by the number of input training examples, or facts. In other words, the number of training examples should be greater than that of trainable weights. In an ideal world, having 10 or more facts for each weight are required. For instance, in a 10–10–1 architecture there are 110 ($= 10 \times 10 + 10 \times 1$) weights, so you should have about 1,100 facts (example data).

C4.0 Backpropagation Network

The backpropagation neural network (BPN) is the most widely used feedforward neural network system. The term backpropagation refers to the training method by which the weights of the network connection are adjusted. The calculations procedure is feedforward, from input layer through hidden layers to output layer. During training, the calculated outputs are compared with the desired values, and then the errors are backpropagated to correct all weight factors. The whole calculation procedure (for a three-layer BPN) is summarized as follows¹²:

1. Randomly assign values between 0 and 1 to weights $W_{i,j}(l)$ for each layer, l . All input-layer thresholds are assigned to zero, i.e. $T_{i,1} = 0$; all hidden- and output-layer thresholds are assigned to one, i.e., $T_{i,3} = 1$.
2. Introduce the input I_i into the neural network, and calculate the output from the first layer according to the equations:

$$x_i = I_i + T_{i,1} \quad (C-6)$$

$$a_{i,1} = f(x_i) \quad (C-7)$$

where $f(.)$ is the transfer function mentioned in the previous section.

3. Knowing the output from the first layer, calculate outputs from the second layer, using the equation:

$$a_{i,2} = f\left(\sum_j^L (W_{j,i}(2)a_{j,1}) + T_{i,2}\right) \quad (C-8)$$

4. Given the output from the second layer, calculate the output from the output-layer, using the equation:

$$a_{i,3} = f\left(\sum_j^L (W_{j,i}(3)a_{j,2}) + T_{i,3}\right) \quad (C-9)$$

$$y_i = a_{i,3} \quad (C-10)$$

Steps 1 to 4 represent the forward activation flow; that is, the given input values I_i move forward in the network, activate the nodes, and produce the actual output values y_i based on the initially assumed values of interconnecting weights, $W_{i,j}(l)$ and internal threshold, $T_{i,l}$. Obviously, the initial calculation will not produce the desired output values (d_i). The next few steps of the backpropagation algorithm represent the backward error flow in which the errors between the desired output d_i and the actual output y_i flow backward through the network and try to find a new set of network parameters ($W_{i,j}(l)$ and T_i).

5. Now backpropagate the error through the network, starting from the output layer and moving backward toward the input layer. Calculate the gradient-descent term ($\delta_{l,3}$) using the equations:

$$x_{i,3} = \sum_j^L (W_{j,i}(3)a_{j,2}) + T_{i,3} \quad (C-11)$$

$$\delta_{i,3} = (d_i - y_i) \frac{\partial f(x_i)}{\partial x_i} \quad (\text{C-12})$$

6. Knowing the output-layer, $\delta_{1,3}$, calculate $\delta_{1,2}$, the gradient-descent term for the hidden layer (layer 2) using the equations:

$$x_{i,2} = \sum_j^L (W_{j,i}(2) a_{j,1}) + T_{i,2} \quad (\text{C-13})$$

$$\delta_{i,2} = \left(\sum_k \delta_{k,3} W_{k,i}(3) \right) \frac{\partial f(x_i)}{\partial x_i} \quad (\text{C-14})$$

7. Knowing the deltas for the hidden and output layers, calculate the weight changes, $\Delta W_{i,j}$, using the equation:

$$\Delta W_{i,j}(l)_{new} = \eta \delta_{i,l} a_{j,l-1} + \alpha \Delta W_{i,j}(l)_{old} \quad (\text{C-15})$$

where η is the learning rate, and α is the momentum coefficient. The momentum term is added to speed up the training rate. The momentum coefficient, α , is restricted to $0 < \alpha < 1$.

8. Knowing the weight changes, update the weights as:

$$W_{i,j}(l)_{new} = W_{i,j}(l)_{old} + \Delta W_{i,j}(l)_{new} \quad (\text{C-16})$$

One iteration has now been completed. This feedforward calculation and error backpropagation procedure is repeated until the sum of errors is less than the specified value. This is the whole learning process for the neural network. The new weight factors are calculated from the old weight factors from the previous training iteration by the following general expression:

$$[W_i]_{new} = [W_i]_{old} + \left[\begin{matrix} \text{learning} \\ \text{rate} \end{matrix} \right] x \left[\begin{matrix} \text{input} \\ \text{term} \end{matrix} \right] x \left[\begin{matrix} \text{gradient-descent} \\ \text{correction term} \end{matrix} \right] x \left[\begin{matrix} \text{momentum} \\ \text{coefficient} \end{matrix} \right] x \left[\begin{matrix} \text{previous} \\ \text{weight change} \end{matrix} \right] \quad (\text{C-17})$$