ABSTRACT

The project is an interactive graphics program that interfaces with a production history database for Legacy Exploration Inc., an oil and gas company. The primary feature of the graphics program is to produce a decline curve on the screen. A decline curve is historical production plotted versus time or cumulative production for an oil or gas well. The program user is able to analyze and extrapolate the curve to estimate future production.

The program is menu driven. The user has the option to view the plot of one or more of the following (versus time or versus cumulative production) on a semilogarithmic scale: gas production, oil production, condensate production, flowing tubing pressure, choke size, major product to water ratio, and minor to major product ratio.

To facilitate analysis of past production, the program performs a least squares fit on past production data points. Using the result of the least squares fit, the program suggests a type of decline (exponential, hyperbolic, or harmonic) that best fits the historical production curve. The user is able to accept or override this suggestion.

The user is able to make his or her own projection by entering appropriate parameters for one of the decline types. As the projected curve is formulated, the screen displays the amount of reserves resulting from the projection. The user has the ability of storing the projected production in the database to use for future analyses.
Table of Contents

1. Introduction
   1.1 Background and Rationale ........................................... 1
   1.2 Goal of the Project .................................................. 4

2. DCA Physical Description
   2.1 DCA System Requirements ............................................ 5
   2.2 DCA User Interaction ................................................ 6
      2.2.1 DCA Start ....................................................... 6
      2.2.2 DCA Menu to Retrieve Data .................................... 6
      2.2.3 Menus and Messages for Interactive Graphics ................. 7
         2.2.3.1 Choose Well Identity Numbers .............................. 7
         2.2.3.2 Main Menu .................................................. 8
         2.2.3.3 Graph Settings Menu .................................... 11
         2.2.3.4 Analysis Menu ............................................. 12
         2.2.3.5 Data Menu ................................................ 13
         2.2.3.6 Graph Type Menu ......................................... 13
         2.2.3.7 Display Menu ............................................. 14
         2.2.3.8 Rate Scale Menu .......................................... 15
         2.2.3.9 Cumulative Production Scale Option ....................... 16
      2.2.4 Menus and Messages to Store Data .............................. 22
      2.2.5 DCA End ....................................................... 22

3. DCA Source Files and Module Descriptions
   3.1 Overview of DCA Structure ......................................... 23
      3.1.1 DOS Level Overview of DCA .................................... 23
      3.1.2 Overview of DCA.EXE ......................................... 26
   3.2 DCA Data Files .................................................... 31
      3.2.1 RBXTRACT.DAT ................................................. 31
      3.2.2 DECLINE.DAT ................................................ 31
      3.2.3 WELLID.DAT .................................................. 32
      3.2.4 FUTURE.DAT .................................................. 32
      3.2.5 NEWFUT.DAT .................................................. 33
      3.2.6 RBSFUT.DAT .................................................. 33
   3.3 Turbo C Executable Programs ...................................... 34
      3.3.1 DCASTART.EXE ............................................... 34
         3.3.1.1 Function Descriptions .................................. 34
         3.3.1.2 Source File ............................................. 34
      3.3.2 GETDATA1.EXE ............................................... 34
         3.3.2.1 Function Descriptions .................................. 34
         3.3.2.2 Source File ............................................. 35
      3.3.3 DCA1.EXE ..................................................... 35
         3.3.3.1 Data Structures ......................................... 35
            3.3.3.1.1 Videostruct ....................................... 35
            3.3.3.1.2 Futfilestruct ..................................... 36
<table>
<thead>
<tr>
<th>3.3.3.1.3 Scalesz</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.3.1.4 Prostruct</td>
<td>37</td>
</tr>
<tr>
<td>3.3.3.1.5 Identstruct</td>
<td>37</td>
</tr>
<tr>
<td>3.3.3.1.6 Wellstruct</td>
<td>38</td>
</tr>
<tr>
<td>3.3.3.1.7 Futurestruct</td>
<td>38</td>
</tr>
<tr>
<td><strong>3.3.3.2 Function Descriptions</strong></td>
<td>39</td>
</tr>
<tr>
<td><strong>3.3.3.3 Source Files</strong></td>
<td></td>
</tr>
<tr>
<td>3.3.3.3.1 DCAMAIN.C</td>
<td>53</td>
</tr>
<tr>
<td>3.3.3.3.2 WELLID.C</td>
<td>53</td>
</tr>
<tr>
<td>3.3.3.3.3 PRCSWELL.C</td>
<td>54</td>
</tr>
<tr>
<td>3.3.3.3.4 GRPHFUNC.C</td>
<td>55</td>
</tr>
<tr>
<td>3.3.3.3.5 PROJMENU.C</td>
<td>55</td>
</tr>
<tr>
<td>3.3.3.3.6 GRPHMENU.C</td>
<td>56</td>
</tr>
<tr>
<td>3.3.3.3.7 ANALMENU.C</td>
<td>56</td>
</tr>
<tr>
<td>3.3.3.3.8 ANALYSIS.C</td>
<td>57</td>
</tr>
<tr>
<td>3.3.3.3.9 ERRORMES.C</td>
<td>59</td>
</tr>
<tr>
<td><strong>3.3.4 FIXFUT.EXE</strong></td>
<td>59</td>
</tr>
<tr>
<td><strong>3.3.4.1 Function Descriptions</strong></td>
<td>60</td>
</tr>
<tr>
<td><strong>3.3.4.2 Source File</strong></td>
<td></td>
</tr>
<tr>
<td><strong>3.3.5 DCASTOP.EXE</strong></td>
<td>60</td>
</tr>
<tr>
<td><strong>3.3.5.1 Function Descriptions</strong></td>
<td>60</td>
</tr>
<tr>
<td><strong>3.3.5.2 Source File</strong></td>
<td></td>
</tr>
</tbody>
</table>

**4. Conclusions**
- **4.1 Project Results** ........................................ 62
- **4.2 Future Work** ........................................... 63

**References** .................................................. 64

**Appendix A - Sample Data** ................................... A1
**Appendix B - Program Listings** ............................... B1
- **DCA.BAT** .................................................. B2
- **A.BAT** ..................................................... B3
- **B.BAT** ..................................................... B4
- **MENU.DCA** .................................................. B5
- **RBXTRACT.EXE** ............................................. B6
- **RBPUTXTR.EXE** ............................................. B7
- **ADDFUT.EXE** ............................................... B8
- **DCASTART.C** ............................................... B9
- **GETDATA1.C** ............................................... B13
- **DCAMAIN.C** ............................................... B20
- **WELLID.C** .................................................. B25
- **PRCSWELL.C** ............................................... B31
- **GRPHFUNC.C** ............................................... B48
- **PROJMENU.C** ............................................... B67
- **GRPHMENU.C** ............................................... B71
- **ANALMENU.C** ............................................... B82
- **ANALYSIS.C** ............................................... B100
- **ERRORMES.C** ............................................... B135
- **DCASTRCT.H** ............................................... B141
- **FIXFUT.C** ................................................. B149
- **DCASTOP.C** ............................................... B154
LIST OF FIGURES

2.1 Data Retrieval Menu ............................................. 6
2.2 Choose Well Identity Numbers Menu ............................ 8
2.3 Well Identity Number Entry Message ........................... 8
2.4 Main Menu ......................................................... 8
2.5 Menu Structure Chart ............................................... 9
2.6 Graph Settings Menu ............................................. 11
2.7 Analysis Menu ..................................................... 12
2.8 Data Item Menu ................................................... 13
2.9 Graph Type Menu .................................................. 14
2.10 Display Menu ..................................................... 14
2.11 Rate Scale Menu for Gas, Oil, and FTP ....................... 15
2.12 Rate Scale Menu for Ratios ..................................... 15
2.13 Change Cumulative Production Scale Menu .................... 16
2.14 Decline Menu ..................................................... 16
3.1 Dos Level Structure Chart ....................................... 24
3.2 DCA1.EXE Structure Chart ...................................... 27
1. INTRODUCTION

1.1 BACKGROUND AND RATIONALE

In Texas, all operators of oil and gas wells must report monthly production volumes to the Railroad Commission. Since production data is tracked monthly, most small, independent operators use this data to plot decline curves. A decline curve is a graph on a semilogarithmic grid representing a well's monthly production plotted versus time or versus cumulative production. Such graphs are termed decline curves because once a well can no longer produce at its allowable limit, its production rate declines from month to month on a regular basis [Nind, p. 32].

Decline curves are used for evaluating well performance because they are convenient, quick, reliable, and are amenable to mathematical analysis. They graphically illustrate when a well is producing below expectations so that a work-over or rehabilitation may be planned. More importantly, they are used to estimate remaining reserves for the well by extrapolating the rate of the decline [Nind, pp.31-32].

Once remaining reserves are known, economic parameters can be applied to the future production values to estimate the amount of future income from the well. This data is used in the preparation of budgets, in portfolios for investors, and in asset statements for loans.

With experimentation, mathematical treatment has been formulated to match the decline curve produced when a well's production rate starts diminishing. The matched curve is then extrapolated into the future. The extrapolated curve is a prediction of future well performance [Nind, p. 32]. Three types of curves are used for matching: exponential, hyperbolic, and harmonic. The curve type depends upon the characteristics of the nominal decline rate [Bradley, p. 40-28]. The nominal decline rate is the negative slope of the curve of the natural logarithm of the production rate versus time [Bradley, p. 40-27].

In exponential decline, the nominal decline rate is constant [Bradley, p. 40-28]. An exponential decline curve is used when the plot of the natural logarithm of production rate versus time is a straight line [Nind, p. 34].

In hyperbolic decline, the nominal decline rate is not constant, but is proportional to a fractional power of the production rate [Bradley, p. 40-28]. The smaller the production rate, the smaller the decline [Nind, p. 38]. An hyperbolic curve will produce a straight line when the reciprocal of the nominal decline rate is plotted versus time [Nind, p. 42]. Also, an hyperbolic curve produces a straight line when the log of the
production rate minus a constant is plotted versus the log of time [Nobles, p. 214].

In harmonic decline, the nominal decline rate is proportional to the production rate [Bradley, p. 40-29]. An harmonic decline curve is used when the natural logarithm of the production rate versus cumulative production produces a straight line [Nindo, p. 40].

In order to reduce time required to plot the logarithm of the monthly production rate, the production rate is plotted on a logarithmic scale. Parameters can be read from this graph and used in computations exactly as if the natural logarithm of the rate were plotted on a unit scale.

At Legacy Exploration Inc., decline curves are manually plotted on a month to month basis. As Legacy grows, it tries to attract more and more investors. Investors request data and reports from the company upon which to base financial decisions. Decline curves are frequently requested items. Due to the low number of personnel, it is not feasible for Legacy to manually plot the data in a timely fashion. They need to automate the plotting of their production histories.

Although existing commercial decline curve analysis programs can extrapolate production curves using exponential, hyperbolic, and harmonic decline, they are either too expensive or too inflexible to meet Legacy's needs. Each of the programs suffers from one or more of the following limitations:

1. Commercial packages do not interface with the existing production history database at Legacy. Legacy does not want to convert their database to match the data tracking system used by commercial packages because they would lose control of their data and the ability to produce ad hoc reports. It is also too expensive to customize a commercial package to interface with Legacy's database.

2. Many commercial packages plot only the major product versus time. In some reservoirs, water production and flowing tubing pressure are as important as the major product in determining a well's reserves.

3. Existing packages do not plot choke size. When extrapolating a curve for a product, one must consider only historical production at the same choke size to correctly evaluate a well's performance; otherwise, reserves will be incorrectly determined.
4. Many packages do not plot a possible minor product whose future production can be a factor in determining total income for the well. If a well produces a minor product, reserve amounts for the minor product are usually included in economic analyses of the well.
1.2 GOAL OF THE PROJECT

The goal of this project was to produce an interactive graphics program that overcomes the aforementioned limitations of current commercial programs. The program that was developed is called DCA for Decline Curve Analysis. DCA overcomes the limitations by having the following features:

1. DCA interfaces with the existing production history database at Legacy. This feature allows personnel at Legacy to produce decline curves in a timely manner.

2. DCA is interactive and easy to use through its user menu facilities.

3. The program gives the user the option to plot one or more of the following on a semilogarithmic scale: gas production, oil production, condensate production, flowing tubing pressure, choke size, major product to water ratio, and minor to major product ratio. The user also has the option of plotting the production data versus time or versus the cumulative production of the major product.

4. DCA performs a least squares fit on past production data points and suggests one of the decline types. The user is able to accept or override this suggestion.

5. The program gives the user the ability to make his or her own projection by entering appropriate parameters for one of the decline types.

6. Whenever the historical production curve is extrapolated as in features 4 and 5, the program displays the resulting reserve amount.

7. The user has the ability of storing the projected production in the database to use for future analyses.
2. DCA PHYSICAL DESCRIPTION

2.1 DCA SYSTEM REQUIREMENTS

DCA runs under MS-DOS version 3.3. The project functions properly on an IBM compatible personal computer with an 80286 processor, 640K of memory, and either an EGA or VGA monitor.

The R:BASE System V, version 1.1 program and the Legacy production history database files should be installed on Drive D in the subdirectory R:BASE. Also, the R:BASE playback files RBXTRACT.EXE, RBFUTXTR.EXE, and ADDFUT.EXE should be in this directory.

The following executable programs and batch files should be on drive C in the subdirectory DCA:

<table>
<thead>
<tr>
<th>Executable Programs</th>
<th>Batch Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCASTART.EXE</td>
<td>DCA.BAT</td>
</tr>
<tr>
<td>GETDATA1.EXE</td>
<td>A.BAT</td>
</tr>
<tr>
<td>DCA1.EXE</td>
<td>B.BAT</td>
</tr>
<tr>
<td>FIXFUT.EXE</td>
<td>MENU.DCA</td>
</tr>
<tr>
<td>DCASTOP.EXE</td>
<td></td>
</tr>
</tbody>
</table>
2.2 DCA USER INTERACTION

DCA is an interactive graphics program that interfaces with a production history database. DCA uses menus and messages to interact with the user. The first message indicates that DCA has been invoked. The first menu is associated with data retrieval from the database. The next series of menus and messages are associated with the graphics portion of the project. They allow the user to choose which wells to process and how data is to be displayed and extrapolated. The last series of menus are associated with storing data in the database. The last message indicates that DCA has ended.

When the user is prompted to enter a response, he should type in his response and then press enter on the keyboard.

2.2.1 DCA START

At the DOS prompt in the DCA subdirectory on drive C, the user types in DCA and presses enter. The screen displays a message about DCA, pauses, and then displays the data retrieval menu.

2.2.2 DCA MENU TO RETRIEVE DATA

The Data Retrieval Menu is shown in Figure 2.1. The menu options are described in this section.

================================================================
DATA RETRIEVAL MENU

A = RETRIEVE HISTORICAL PRODUCTION DATA
   AND FUTURE PRODUCTION DATA

B = RETRIEVE ONLY FUTURE PRODUCTION DATA

Choose option A if there have been changes made to the production history database because DCA historical production data files have to be updated.

Both options update DCA data files associated with future data.

TYPE IN THE LETTER FOR THE OPTION YOU WISH TO EXECUTE
   AND PRESS ENTER.

================================================================

Figure 2.1

Option A updates data files associated with historical production. Updated data is retrieved from the database. The
R:BASE FileGateway menus are displayed on the screen but no interaction is required from the user. Preprocessing is performed on the historical database files. Then option A invokes option B.

Option B updates data files associated with future production data. Future data is retrieved from the database. The R:BASE FileGateway menus are displayed on the screen but no interaction is required from the user. Then option B invokes the interactive graphics portion of the project (Section 2.2.3).

2.2.3 MENUS AND MESSAGES FOR INTERACTIVE GRAPHICS

The menus and messages for interactive graphics are implemented through functions in the executable file DCA1.EXE and are described in this section.

When processing an oil well, "oil/condensate" production refers to oil production. "Ratio" refers to cubic feet of gas per barrel of oil (Cf/Bbl). "Water" refers to barrels of oil per barrel of water (Bbl/Bbl). The cumulative production graph type plots monthly production rates versus cumulative oil production.

When processing a gas well, "oil/condensate" refers to condensate production. "Ratio" refers to barrels of oil per thousand cubic feet of gas (Bbl/Mcf). "Water" refers to cubic feet of gas per barrel of water (Cf/Bbl). The cumulative production graph type plots monthly production rates versus cumulative gas production.

2.2.3.1 Choose Well Identity Numbers

The options available to the user for choosing wells to process from the database are shown in Figure 2.2.

Option A causes all wells in the database to be processed in sequential order by well identity number. The well with lowest identity number is displayed first on the screen with the interactive graphics Main Menu (Section 2.2.3.2).

If option B is chosen, then the message in Figure 2.3 is displayed. The user is not required to enter well identity numbers in sequential order, but well plots will be displayed in sequential order by well identity number. If the user enters an invalid well identity number, an error message is displayed and the program continues processing. When the user is finished entering well identity numbers, he should enter zero. The program then displays the interactive graphics Main Menu (Section 2.2.3.2).
CHOOSE WELL IDENTITY NUMBERS

A - All wells in the database

B - Enter well identity numbers for specific wells

Choose option A if you want to have the program produce a decline curve for all wells in the database. (Press A and enter).

Choose option B if you want to view a decline curve for only specific wells in the database. (Press B and enter).

Figure 2.2

Type in the well identity number and press enter. Type in 0 (zero) and press enter when finished entering well identity numbers.

Well identity number:

Figure 2.3

2.2.3.2 MAIN MENU

After the user has selected the wells to process, the plot for the first well is displayed on the screen. The interactive graphics Main Menu (Figure 2.4) is at the top of the screen. The relationship between the Main Menu and its submenus is shown in Figure 2.5. The menu options are described in this section.

MAIN MENU

G - Graph Settings Menu  
A - Analysis Menu  
O - Output Menu  
S - Save Future Projection & Display Next Well  
X - Exit program

Type in the letter of the option you wish to execute:

Figure 2.4

Option G takes the user to the Graph Settings Menu (Section 2.2.3.3). These menu options allow the user to change the display of the decline curve on the screen.

Option A takes the user to the Analysis Menu (Section 2.2.3.4). These menu options allow the user to analyze and extrapolate the decline curve using decline curve equations.

Option O takes the user to the Output Menu. The Output Menu options produce a copy of the decline curve on a printer or plotter and will be implemented in a future version of the
program. The options are not presently implemented due to a lack of available resources.

Option S causes future production data to be saved in a data file for inclusion in the production history database. If the user does not want to save future production data, then he should invoke the Analysis Menu (Section 2.2.3.4) and choose the erase option. The user should erase all future projected production for all data items. Then, the user can return to the Main Menu and choose option S to continue processing.

If there are more wells to process, then the next well number is displayed upon the screen. If there are no more wells, the interactive graphics portion of DCA terminates. Then DCA replaces future production data in the database by invoking the R:BASE FileGateway program. The user is prompted to exit out of the FileGateway program (Section 2.2.4).

Option X terminates the interactive graphics portion of DCA. Then DCA replaces future production data in the database by invoking the R:BASE FileGateway program. The user is prompted to exit out of the FileGateway program (Section 2.2.4).

2.2.3.3 GRAPH SETTINGS MENU

The Graph Settings Menu is shown in Figure 2.6. The menu options are described in this section.

```
GRAPH SETTINGS MENU
G - Graph Type Menu     D - Display Menu     R - Rate Scale Menu
C - Cumulative Production Scale Menu  P - Parent Menu
Type in the letter of the option you wish to execute:

Figure 2.6
```

Option G allows the user to change the type of graph displayed (Section 2.2.3.6).

Option D allows the user to toggle the display of a data item on and off (Section 2.2.3.7).

Option R allows the user to change the rate scale of the y-axis for a data item. First, the Data Menu is displayed, and the user is prompted to choose a data item to process (Section 2.2.3.5). Once chosen, the user is prompted for the scale change to be made to the chosen data item (Section 2.2.3.8). The user is then returned to the Data Menu to optionally change the scale for another data item.
Option C allows the user to change the scale of the x-axis when plotting versus cumulative production (Section 2.2.3.9).

Option P returns to the Main Menu (Section 2.2.3.2).

2.2.3.4 ANALYSIS MENU

The Analysis Menu is shown in Figure 2.7. Its menu options are described in this section.

=================================================================
ANALYSIS MENU
C - Curve Fit Calculations  D - Decline Curve Equations
   I - Input via Mouse
   E - Erase Future Projection  P - Parent Menu
Type in the letter of the option you wish to execute:
=================================================================

Figure 2.7

Option C performs curve fit calculations on a data item. First, the Data Menu (Section 2.2.3.5) is displayed, and the user is prompted to choose a data item to process. The curve fit is performed on the data item. Results are displayed on the screen (see Section 2.2.3.11). The user has the option of using the fit to extrapolate data. If the user accepts the fit, he is prompted for parameters (Section 2.2.3.11). Results are calculated and displayed. The user is then returned to the Data Menu to optionally perform a curve fit on another data item.

Option D allows the user to project future production using either constant rate, exponential decline, hyperbolic decline, or harmonic decline. First, the data menu (Section 2.2.3.5) is displayed, and the user is prompted to choose a data item to process. Then the Decline Menu (Section 2.2.3.10) is displayed. The user is prompted to select a decline type. Once a decline type has been chosen, the user is prompted for parameters so that reserves can be determined (Section 2.2.3.10). Results are calculated and displayed. The program then returns to the Decline Menu so that the user can continue processing the chosen data item.

Option I takes the user to the Input Via Mouse Menu which allows the user to draw on the graph any production scheme desired via a mouse device. Option I has not been implemented due to a lack of available resources.

Option E allows the user to erase future projected production. First the data menu (Section 2.2.3.5) is displayed, and the user is prompted to choose a data item to process. Once a data item has been chosen, parameters are obtained from the user (Section 2.2.3.12). Results are calculated and displayed. The
user is then returned to the Data Menu to optionally erase future production for another data item.

Option P returns to the Main Menu (Section 2.2.3.2).

2.2.3.5 DATA MENU

The Data Menu is shown in Figure 2.8. Its menu options are described in this section.

=================================
Choose a Data Item to Process:
  O - Oil/Condensate    G - Gas    R - Ratio    W - Water
  F - FTP                S - Refresh Screen    P - Parent Menu

Type in the letter of the option you wish to execute:
=================================

Figure 2.8

Option O selects processing to be performed on oil or condensate production.

Option G selects processing to be performed on gas production.

Option R selects processing to be performed on the minor to major product ratio.

Option W selects processing to be performed on the major product to water production ratio.

Option F selects processing to be performed on flowing tubing pressure.

Options S refreshes the screen.

Option P returns to the menu that invoked the Data Menu.

Types of processing performed on a data item are rate scale change, curve fit calculations, data projection using decline curve equations, and erasing future production. These processes call the data menu and program control returns to the process that invoked the Data Menu after a data item has been selected for processing.

2.2.3.6 GRAPH TYPE MENU

The Graph Type Menu is shown in Figure 2.9. The menu options are described in this section.
GRAPH TYPE MENU

Plot Monthly Production Versus:  T - Time or  C - Cumulative Production

P - Parent Menu

Type in the letter of the option you wish to execute:

Figure 2.9

Option T redraws the graph with monthly production plotted versus time. The time unit is months.

Option C redraws the graph with monthly production plotted versus cumulative production of the major product. If the user does not like the cumulative production scale, he can change the scale through option C on the Graph Settings Menu (Section 2.2.3.3).

Option P returns to the Graph Settings Menu (Section 2.2.3.3).

Note that when the graph type is changed, subsequent plots for other wells will be of the new graph type.

2.2.3.7 DISPLAY MENU

The Display Menu is shown in Figure 2.10. Its menu options are described in this section.

To Turn the Display of a Data Type ON and OFF,
TOGGLE:  O - Oil/Condensate  G - Gas  R - Ratio  W - Water  C - Choke  F - FTP  S - Refresh Screen  P - Parent Menu

Type in the letter of the option you wish to execute:

Figure 2.10

Option O toggles the display of the oil or condensate production on and off. If oil production is currently displayed, choosing option O causes oil production to disappear. If oil production exists and it is not currently on the screen, choosing option O causes oil production to appear.

Options G, R, W, C, and F work similarly for the other data items. Gas is for gas production. Ratio is for the minor to major product ratio. Water is for the major product to water production ratio. Choke is for choke size. FTP is for flowing tubing pressure.
Option S refreshes the screen. In some cases, the scale units along the y-axis will not appear. Choosing option S causes all units for displayed data items to appear.

Option P returns to the Graph Settings Menu (Section 2.2.3.3).

NOTE: When the display of a data item is toggled off, subsequent plots for other wells will not plot the data item until the display of the data item is toggled on.

2.2.3.8 RATE SCALE MENU

These menus allow the user to change the scale of the y-axis for the different data items. The menu that appears will depend upon the data item that was chosen for processing. The Rate Scale Menu for the minor to major product ratio and the major product to water production ratio is shown in Figure 2.11. The Rate Scale Menu for oil/condensate production, gas production, and flowing tubing pressure is shown in Figure 2.12.

=================================================================
Choose Maximum Value for Scale (Bbls/Mcf, Cf/Bbl, or Bbl/Mcf):
A - 0.01  B - 0.1  C - 1.0  D - 10.0  E - 100.0  F - 1000
  G - 10,000
  P - Parent Menu

Type in the letter of the option you wish to execute:
=================================================================

Figure 2.11

=================================================================
Choose Maximum Value for Scale (Bbls/Month, Mcf/Month, or
  Psi/Month):
A - 1,000  B - 10,000  C - 100,000  D - 1,000,000
  E - 10,000,000
  P - Parent Menu

Type in the letter of the option you wish to execute:
=================================================================

Figure 2.12

The menu options represent the maximum scale value for monthly production rates on the semilogarithmic grid. When a menu option is invoked, the graph is redrawn with the new scale size. The user is then returned to the Data Menu (Section 2.2.3.5) to choose another data item to process.

Option P returns to the Data Menu (Section 2.2.3.5) to allow the user to choose another data item to process.
2.2.3.9 CUMULATIVE PRODUCTION SCALE OPTION

The Cumulative Production Scale Option is shown in Figure 2.13. The option is described in this section.

Change Cumulative Production Scale
Scale size must be at least 10 MMCF or 10 MBBLS
and a multiple of the number 10.
Enter new maximum value for cumulative production scale:

Figure 2.13

This option allows the user to change the maximum scale value for cumulative production on the x-axis if he does not like the size chosen by the program.

The units along the x-axis for cumulative gas production are millions of cubic feet. The units for cumulative oil production are thousands of barrels. User entered maximum scale values will also be in these units. The user entered value must be greater than 10 and a multiple of the number 10. If the user does not enter a value that meets these specifications, an error message is displayed. When a valid value for the scale has been entered, the graph is redrawn with the new scale size.

2.2.3.10 DECLINE MENU

The Decline Menu is shown in Figure 2.14. The menu options are described in this section.

Decline Curve Calculations
C - Constant Rate  E - Exponential  H - Hyperbolic  A - Harmonic
R - Erase Future Projection  S - Refresh Screen  P - Parent Menu
Type in the letter of the option you wish to execute:

Figure 2.14

Option C allows the user to project data at a constant rate of production. The user is prompted for the following parameters: measurement unit for production rate, constant rate of production, beginning date of when to start constant rate of production, and an ending date of when to stop projection of constant rate of production. See Section 2.2.3.13 for more information about input parameters. After all parameters are input, results are calculated and displayed on the screen. The program returns to the Decline Menu for further processing on the data item.
Option E allows the user to project data using exponential decline curve equations (Section 2.2.3.15). The user is prompted for the following parameters: beginning date of when to start exponential decline, measurement units, initial rate of production, effective decline rate, and an end condition. See Section 2.2.3.13 for more information about input parameters. After all parameters are input, results are calculated and displayed on the screen. The program returns to the Decline Menu for further processing on the data item.

Option H allows the user to project data using hyperbolic decline curve equations (Section 2.2.3.15). The user is prompted for the following parameters: beginning date of when to start hyperbolic decline, measurement units, initial production rate, initial effective decline rate, hyperbolic exponent, and an end condition. See Section 2.2.3.13 for more information about input parameters. After all parameters are input, results are calculated and displayed on the screen. The program returns to the Decline Menu for further processing on the data item.

Option A allows the user to project data using harmonic decline curve equations (Section 2.2.3.15). The user is prompted for the following parameters: beginning date of when to start harmonic decline, measurement units, initial production rate, initial effective decline rate, and an end condition. See Section 2.2.3.13 for more information about input parameters. After all parameters are input, results are calculated and displayed on the screen. The program returns to the Decline Menu for further processing on the data item.

Option R allows the user to erase future projected production. The user is prompted for an erasure date (see Section 2.2.3.12). After the erasure date is entered, results are calculated and displayed. The program returns to the Decline Menu for further processing on the data item.

Option S refreshes the graph on the screen.

Option P returns to the Analysis Menu (Section 2.2.3.4).

NOTE: When production for the major and minor product are projected, the minor to major product ratio is also projected. If the ratio and the major product are projected, then the minor product is projected. If future production exists for the ratio, minor product, and major product and a change is made to the major product, the user is prompted to change either the ratio or minor product so that data can remain consistent.

NOTE: Future water production is not calculated unless the major product and major product to water ratio are both projected into the future.
2.2.3.11 CURVE FIT INTERACTION

DCA tries to formulate an exponential, hyperbolic, or harmonic curve that best fits past production. DCA uses least squares methods on equations 3, 9, 11, and 15 in Section 2.2.3.15 to find an exponential, hyperbolic, or harmonic curve. The curve that has the smallest percent standard deviation from actual historical data is chosen as the fit. The largest allowable percent standard deviation is 10,000%.

If the program is able to formulate a fit, a message is displayed describing the curve and its parameters. Then the user is asked if he would like to project future production using the parameters. If the user chooses to project production, then he is asked for an end condition (Section 2.2.3.13). Results are calculated and displayed. The program returns to the Data Menu to allow the user to choose another data item for processing.

If the program is unable to formulate a fit, a message is displayed that no fit was found. Then the program returns to the Data Menu to allow the user to choose another data item for processing.

2.2.3.12 ERASE FUTURE PRODUCTION INTERACTION

DCA prompts the user to enter an erasure date. Monthly future production data values for the chosen data item are erased starting at the erasure date. The date should be greater than the last production date for the chosen data item. The last production date is displayed on the screen. The erasure date should be entered in the form of MM/YYYY. If an incorrect date is entered, an error message is displayed.

2.2.3.13 PARAMETER SPECIFICATIONS

Measurement unit options are listed in Section 2.2.3.14. DCA converts all production rates to monthly rates for use in decline curve equations.

An end condition signals when to stop projecting future production. Future production is stopped either when a production rate is reached or when an ending date is reached.

Production rates should be greater than zero. Initial production rate is the instantaneous production rate at the beginning of the decline period (theoretical time zero).

Beginning dates should be later in time than the date of last production and later in time than the last date of future production for a data item. If a date is used as an end condi-
tion, then the ending date should be later than the beginning date. The year in both dates should be less than 3000. Dates should be entered in the form of MM/YYYY.

Decline rate percentages should be entered in decimal format. The effective decline rate is the drop in production rate from rate\(_1\) to rate\(_2\) divided by rate\(_1\) for a period of time such as 1 day, 1 month, or 1 year. The initial effective decline rate is the instantaneous effective decline rate at the beginning of the decline period (theoretical time zero).

The hyperbolic exponent is the exponent \((n)\) used in hyperbolic equations (Section 2.2.3.15). The exponent must be greater than 0, not equal to 1, and less than 10. A value of 0 should not be used for the hyperbolic exponent because this situation indicates exponential decline. For this case, the user should use exponential decline curve equations to project production. A value of 1 should not be used for the hyperbolic exponent because this situation indicates harmonic decline. For this case, the user should use harmonic decline curve equations to project production. A value greater than 10 for the hyperbolic exponent is unrealistic.

If input parameters do not meet the above specifications, an appropriate error message is displayed.

2.2.3.14 MEASUREMENT UNITS

Before entering a parameter such as a production rate, the user must identify the units of the parameter. The allowable units on parameters for different data items are:

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Allowable Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil Production Rate</td>
<td>Barrels of Oil/Day</td>
</tr>
<tr>
<td></td>
<td>Barrels of Oil/Month</td>
</tr>
<tr>
<td></td>
<td>Barrels of Oil/Year</td>
</tr>
<tr>
<td>Gas Production Rate</td>
<td>Thousands of Cubic Feet/Day</td>
</tr>
<tr>
<td></td>
<td>Thousands of Cubic Feet/Month</td>
</tr>
<tr>
<td></td>
<td>Thousands of Cubic Feet/Year</td>
</tr>
<tr>
<td>Oil Production to Water Production Ratio</td>
<td>Barrels of Oil/Barrel of Water</td>
</tr>
<tr>
<td>Gas Production to Water Production Ratio</td>
<td>Cubic Feet of Gas/Barrel of Water</td>
</tr>
<tr>
<td>Condensate Production to Gas Production Ratio</td>
<td>Barrels of Condensate/Thousand Cubic Feet of Gas</td>
</tr>
</tbody>
</table>
### Data Item | Allowable Units
--- | ---
Gas Production to Oil Production Ratio | Cubic Feet of Gas/Barrel of Oil
Flowing Tubing Pressure | Pounds per Square Inch/Month
Effective Decline Rate | Percent Decline/Day, Percent Decline/Month, Percent Decline/Year

#### 2.2.3.15 EQUATIONS

The following equations are used in projecting future production and in curve fitting historical production using decline curve equations:

\[ d = \text{effective decline rate} \]
\[ d_0 = \text{initial effective decline rate} \]
\[ b = \text{nominal decline rate} \]
\[ b_0 = \text{initial nominal decline rate} \]
\[ q_0 = \text{initial production rate} \]
\[ q_i = \text{production rate at month } i \]
\[ q_f = \text{final production rate} \]
\[ Q = \text{cumulative production} \]
\[ t = \text{time in months} \]
\[ n = \text{hyperbolic exponent} \]
\[ n_{op} = \text{number of points} \]
\[ c_1, c_2, c_3 = \text{constants} \]
\[ \ln = \text{natural logarithm} \]
\[ x = \text{coordinate on } x\text{-axis} \]
\[ y = \text{coordinate on } y\text{-axis} \]
\[ x_{avg} = \text{average value of all } x \text{ coordinates} \]
\[ y_{avg} = \text{average value of all } y \text{ coordinates} \]
\[ m = \text{slope of line} \]
\[ c_4 = \text{y intercept in equation of a line} \]
\[ \exp = \text{exponential} \]
\[ \text{ste} = \text{percent standard error} \]

In all equations:

\[ d = \frac{(q_i - q_f)}{q_i} \quad [\text{Bradley, p.40-27}] \]

Exponential decline curve equations:

1. \[ b = -\ln(1 - d) \quad [\text{Bradley, p.40-29}] \]
2. \[ q_i = q_0 \times \exp(-b \times t) \quad [\text{Nind, p. 34}] \]
3. \( \ln(q_i) = \ln(q_0) - b \times t \) \quad [\text{Nind, p. 34}]
4. \( q_i/q_0 = q_2/q_1 = \ldots = \exp(-b) \) \quad [\text{Nind, p. 34}]
5. \( t = (-1/b) \times \ln(q_f/q_0) \) \quad [\text{derived from equation 2}]

**Hyperbolic decline curve equations:**

6. \( b_0 = 1/n \times [(1 - d_0)^n - 1] \) \quad [\text{Bradley, p.40-29}]
7. \( q_i = q_0 \times [(1 + n*b_0*t)^{-1/n}] \) \quad [\text{Bradley, p.40-28}]
8. \( t = \left[ \frac{(q_f/q_0)^n - 1}{(n*b_0)} \right] \) \quad [\text{derived from equation 7}]
9. \( 1/b = 1/b_0 + nt \) \quad [\text{Nind, p. 42}]
10. \( q_i = c_1 \times t^{c_2} + c_3 \) \quad [\text{Nobles, p. 214}]
11. \( \ln(q_i - c_3) = \ln(c_1) + c_2 \times \ln(t) \) \quad [\text{Nobles, p. 214}]

**Harmonic decline curve equations:**

12. \( b_0 = d_0/(1 - d_0) \) \quad [\text{Bradley, p.40-29}]
13. \( q_i = q_0 / (1 + b_0 \times t) \) \quad [\text{Bradley, p.40-29}]
14. \( t = \frac{[(q_f/q_0) - 1]}{b_0} \) \quad [\text{derived from equation 13}]
15. \( \ln(q_i) = \ln(q_0) - (b_0/q_0) * Q \) \quad [\text{Nind, p. 40}]

**Percent Standard Error:**

16. \[
\text{ste} = \left[ \frac{\sum \left( \frac{\text{actual point-fit point}}{\text{fit point}} \right)^2}{\text{nop} - 2} \right]^{1/2} \times 100
\]

[\text{derived from Hoel, p. 230}]
Equations used in Least Squares Method:

To determine the equation of a line \( y = mx + c_4 \) using the method of least squares:

\[
\sum x_i y_i - \text{nop} \times \text{xavg} \times \text{yavg}
\]

\[
17. m = \frac{\sum x_i y_i - \text{nop} \times \text{xavg} \times \text{yavg}}{\sum x_i^2 - \text{nop} \times \text{xavg}^2}
\]  

[Hoel, p. 227]

\[
18. c_4 = \text{yavg} - m \times \text{xavg} 
\]  

[Hoel, p. 226]

### 2.2.4 MENUS AND MESSAGES TO STORE DATA

Updated future production data is replaced in the database using an R:BASE "Exec" playback file. The playback file contains responses to menus and messages in the R:BASE FileGateway program. DCA invokes the FileGateway program using the playback file. However, the playback file does not completely take the user out of the FileGateway program due to a limitation of R:BASE.

The user should "press any key to continue" when the FileGateway program prompts him to do so. When the Import Menu is reached, the user should select option 9 to return to the FileGateway Main Menu. From the FileGateway Main Menu, the user should select option 4 to exit FileGateway. The user should now be out of FileGateway. DCA is ready to terminate (Section 2.2.5).

### 2.2.5 DCA END

The screen displays a message that DCA has ended and pauses. The screen is cleared. The user is in DOS in the DCA subdirectory on drive C.
3. DCA SOURCE FILES AND MODULE DESCRIPTIONS

3.1 OVERVIEW OF DCA STRUCTURE

DCA is an interactive graphics program that interfaces with a production history database. The interface with the database is accomplished by using the R:BASE FileGateway program, R:BASE "Exec" playback files, and Turbo C executable programs. The R:BASE playback files are RBXTRACT.EXE, RBFUTXTR.EXE, and ADDFUT.EXE. The Turbo C executable programs involved in the database interface are GETDATA1.EXE and FIXFUT.EXE. The interactive graphics portion of the program is accomplished through a Turbo C executable program, DCA1.EXE. Also, there are two other Turbo C executable programs which display when the project has started and ended. These programs are DCASTART.EXE and DCASTOP.EXE. In order to achieve cohesion between the database interface and the Turbo C executable programs, DOS batch files are implemented.

3.1.1 DOS LEVEL OVERVIEW OF DCA

Three DOS batch files have been implemented so that the project runs as one program to the user. The batch files are DCA.BAT, A.BAT, and B.BAT. A DOS level structure chart is shown in Figure 3.1.

The user starts the program by executing DCA.BAT when he types in DCA at the DOS prompt (Section 2.2.1) in the subdirectory DCA on drive C. DCA.BAT causes the Turbo C executable program DCASTART.EXE to run. This program displays the DCA start up message. After DCASTART.EXE terminates, DCA.BAT displays the information in the file MENU.DCA on the screen. This information is the Data Retrieval Menu described in Section 2.2.2. DCA.BAT has now terminated. The user is at the DOS prompt with the Data Retrieval Menu on the screen.

The user response at the DOS prompt determines whether A.BAT or B.BAT is executed. If the user types in A, then A.BAT is executed. If the user types in B, then B.BAT is executed. A.BAT will cause B.BAT to be executed, so B.BAT is executed in both cases.

The purpose of A.BAT is to update DCA data files associated with historical production. A.BAT need only be executed when historical production has changed in the database. A.BAT deletes the three data files associated with historical production. These data files are RBXTRACT.DAT, DECLINE.DAT, and WELLID.DAT. Then A.BAT switches to the directory that contains the R:BASE program and the Legacy database files. A.BAT causes the R:BASE FileGateway program to be executed using the R:BASE "Exec"
playback file RBXTRACT.EXE. This playback file contains the keystrokes for responses to the FileGateway menus and prompts. No input is needed from the user. (See page 1-21 of the R:BASE System V User's Manual for more information on playback files.) The playback file causes all historical production data to be written to the file RBXTRACT.DAT. Once the data retrieval process is completed, the playback file exits the FileGateway program. Program control returns to A.BAT.

The batch file switches back to the subdirectory DCA on drive C and executes the Turbo C program GETDATA1.EXE. The purpose of this program is to do preprocessing on the file RBXTRACT.DAT before the interactive graphics portion of the project is executed. Input to GETDATA1.EXE is RBXTRACT.DAT. GETDATA1.EXE rewrites the data in RBXTRACT.DAT to the file DECLINE.DAT with redundant data deleted. GETDATA1.EXE writes well identity numbers to the file WELLID.DAT. When GETDATA1.EXE terminates, A.BAT will cause B.BAT to be executed.

The purpose of B.BAT is to update DCA data files associated with future production, execute the interactive graphics portion of the project, and store updated future production in the database. First, B.BAT deletes the three data files associated with future production. These data files are FUTURE.DAT, NEWFUT.DAT, and RBSFUT.DAT. Then B.BAT switches to the directory that contains the R:BASE program and the Legacy database files. B.BAT causes the R:BASE FileGateway program to be executed using the R:BASE "Exec” playback file RBFUTXTR.EXE. This playback file causes all future production data to be written to the file FUTURE.DAT. Once the data retrieval process is completed, the playback file exits the FileGateway program. Program control returns to B.BAT. The batch file switches back to the subdirectory DCA on drive C and executes the Turbo C program DCA1.EXE.

DCA1.EXE is the interactive graphics portion of the project (Section 2.2.3). Input to this program is DECLINE.DAT, WELLID.DAT, and FUTURE.DAT. Also, the user choices on how data is to be extrapolated is input. Output from this program is new future data for each well processed by DCA1.EXE. This output is written to the file NEWFUT.DAT whenever the user chooses option S from the interactive graphics Main Menu (Section 2.2.3.2). DCA1.EXE terminates after all wells are processed or when the user chooses to exit the program. When DCA1.EXE terminates, program control returns to B.BAT.

Then B.BAT causes the Turbo C program FIXFUT.EXE to execute. FIXFUT.EXE merges the file FUTURE.DAT and NEWFUT.DAT into a single file RBSFUT.DAT. The reason for the merge is FUTURE.DAT may contain future data for wells that were not processed when DCA1.EXE was executed. Input to FIXFUT.EXE is FUTURE.DAT and NEWFUT.DAT. Output from FIXFUT.EXE is RBSFUT.DAT. After
FIXFUT.EXE terminates, B.BAT switches to the directory that contains the R:BASE FileGateway program. The batch file invokes FileGateway using the playback file ADDFUT.EXE. This playback file causes the table in the database that contains future production to be replaced by the data in the file RBSFUT.DAT. Input is needed from the user because the playback file will not completely exit out of the FileGateway program.

Once FileGateway is terminated, B.BAT switches back to the DCA subdirectory on drive C. Then B.BAT causes DCASTOP.EXE to execute. DCASTOP.EXE displays a message on the screen that the program is ending, pauses, and terminates. After DCASTOP.EXE terminates, B.BAT terminates. The user is left in DOS in the DCA subdirectory on drive C.

3.1.2 OVERVIEW OF DCA1.EXE

DCA1.EXE is the interactive graphics portion of the project. DCA1.EXE is compiled from the project file DCA1.PRIJ. A structure chart for DCA1.EXE is shown in Figure 3.2. DCA1.PRIJ consists of the files DCAMAIN.C, WELLID.C, PRCSWELL.C, GRPHFUNC.C, PROJMENU.C, GRPHMENU.C, ANALMENU.C, ANALYSIS.C, and ERRORMES.C. These files and the functions they contain are discussed in Section 3.3.3. The file DCASTRUCT.H contains programmer defined data structures and is discussed in Section 3.3.3.1.

DCAMAIN.C contains the main function for DCA1.PRIJ. Main() begins by branching to the function getwellid() to get the user's selection of wells to process (Section 2.2.3.1). After the user indicates which wells to process, program control returns to main(). If the user chooses to process all wells, then all wells are processed by a call to the function processallwells(), otherwise a call is made to the function processfewwells().

Processallwells() and processfewwells() read data for each well in a loop. After data for one well is read, a branch is made to the function prepgraph(). This function calls many other functions to process the well data so that all data necessary to plot the well's monthly production on a semilogarithmic scale is available. Then prepgraph() calls grphvstime() or grphvscum() depending upon the graph type. The initial graph type is to plot monthly production versus time. Grphvstime() calls all functions necessary to display on the screen a plot of monthly production versus time. Grphvscum() calls all functions necessary to display a plot of monthly production versus cumulative production. Then prepgraph() calls the function startopt().

Startopt() implements the interactive Main Menu (Section 2.2.3.2). This menu, its submenus, and their options are all implemented through specific functions.
Structure Chart
DCAI.EXE

Get user's choice of wells to process getwellid()

Initialize Graphics System init_graphics()

Open Data Files

Process Wells processfewwells() or processallwells()

Exit Program exitproj()

Check Video Monitor Type

Split Screen into Text and Graphics Areas graphscrn()

Read Historical Well Data from Data File

Prepare Data for Graphing & Analysis prepgraph()

Display Choose Well Identity Number Menu chooseidmenu()

Get Well Identity Numbers from User enterwellid()

Check for Duplicate Numbers checkdups()

Check validity of Well Identity Number checkwellid()

Get Future Production Data getfut()

Calculate Grid Sizes calgridO; calatgridO; calcumgridO

Get Y and Y Coordinates tresdatalogO; graphycum()

Display Curve on Screen graphystimeO; graphycum()

Start Program Menu Facilities startopt()

Save New Future Data savfutdat()

Get Y Coordinate for Historical Prod. logplot1O

Get Y Coordinate for Future Prod. logplot2O

Calculate Y Coordinates for Choke Data

Calculate X Coordinates when Plotting vs. Time

Calculate X Coor. for Cumulative Prod. xaxiscum()

Draw Choke Grid drawchkgrdO; drawechkgrdO

Draw 3 Cycle Semilogarithmic Grid gridavgO

Display X Axis Labels

Plot Historical & Future Production prodvsxO; futvsxO

Display Well Identity Info. & Y Axis Scale vgolabelsO; egalabelsO

Display Production and Reserve Amounts disresults()

1 see Figure 3.2b

Figure 3.2a
Figure 3.2c
When a user is finished processing a well (option S in Section 2.2.3.2), program control returns to prepgraph(). Prepgraph() calls the function savfutdat() to save new future production data. Then program control returns to either processallwells() or processfewwells() which process the next well. When there are no more wells to process, program control returns to the function main(). Main() calls exitproj() which closes all files, clears the screen, and exits the program.

If the user chooses to exit the project (option X in Section 2.2.3.2) and discontinue processing well data, then the branch to exitproj() is made from within startopt(). The program terminates without returning to prepgraph().
3.2 DCA DATA FILES

Six data files are used by DCA to ensure proper data transferral between the database and the project. These files are RBXTRACT.DAT, DECLINE.DAT, WELLID.DAT, FUTURE.DAT, NEWFUT.DAT and RBSFUT.DAT. The data files are described in this section.

3.2.1 RBXTRACT.DAT

RBXTRACT.DAT contains data extracted from the database. The file contains well identity information and historical monthly production. The data is in ASCII fixed field format and in sequential order by well identity number. The following table describes the data in each line of the file.

<table>
<thead>
<tr>
<th>Data Description</th>
<th>Field Type</th>
<th>Start Position</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well Identity Number</td>
<td>Integer</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Major Product</td>
<td>Character</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>Production Date</td>
<td>Character</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>Monthly Gas Production</td>
<td>Integer</td>
<td>23</td>
<td>10</td>
</tr>
<tr>
<td>Monthly Oil Production</td>
<td>Integer</td>
<td>33</td>
<td>10</td>
</tr>
<tr>
<td>Monthly Water Prod.</td>
<td>Integer</td>
<td>43</td>
<td>10</td>
</tr>
<tr>
<td>Choke Size</td>
<td>Real</td>
<td>53</td>
<td>16</td>
</tr>
<tr>
<td>Flowing Tubing Pressure</td>
<td>Integer</td>
<td>69</td>
<td>10</td>
</tr>
<tr>
<td>Well Name</td>
<td>Character</td>
<td>79</td>
<td>36</td>
</tr>
<tr>
<td>Well Number</td>
<td>Character</td>
<td>115</td>
<td>10</td>
</tr>
<tr>
<td>Field</td>
<td>Character</td>
<td>125</td>
<td>20</td>
</tr>
<tr>
<td>Reservoir</td>
<td>Character</td>
<td>145</td>
<td>20</td>
</tr>
<tr>
<td>Perforations</td>
<td>Character</td>
<td>165</td>
<td>14</td>
</tr>
<tr>
<td>Operator Name</td>
<td>Character</td>
<td>179</td>
<td>25</td>
</tr>
</tbody>
</table>

3.2.2 DECLINE.DAT

GETDATA1.EXE produces DECLINE.DAT from RBXTRACT.DAT. Data in DECLINE.DAT is in sequential order by well identity number. The first line of data for a well is in ASCII fixed field format except for the first two pieces of information. These data items are integers which are delimitied from the rest of the data in the line by a space. The following table describes the information in this line.

<table>
<thead>
<tr>
<th>Data Description</th>
<th>Field Type</th>
<th>Start Position</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well Identity Number</td>
<td>Integer</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Major Product Type</td>
<td>Integer</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Well Name</td>
<td>Character</td>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td>Well Number</td>
<td>Character</td>
<td>41</td>
<td>10</td>
</tr>
<tr>
<td>Field</td>
<td>Character</td>
<td>51</td>
<td>20</td>
</tr>
</tbody>
</table>
For each month of production for a well, there is a line of data in ASCII delimited format. The delimiter is the space. Each line of data contains the following information: well identity number, month in date of production, year in date of production, monthly gas production, monthly oil production, monthly water production, major product to water ratio, flowing tubing pressure, choke size, minor to major product ratio, and monthly cumulative production for the major product. The last line of data for a well is in ASCII delimited format. The delimiter is the space. The line contains well identity number, total number of months of production, year production started, maximum monthly gas production, maximum monthly oil production, maximum water production, maximum major product to water ratio, maximum flowing tubing pressure, maximum choke size, maximum minor to major product ratio, and total cumulative production of the major product.

3.2.3 WELLID.DAT

GETDATA1.EXE also produces WELLID.DAT from RBXTRACT.DAT. WELLID.DAT contains all well identity numbers found in RBXTRACT.DAT. The numbers are written in sequential order in WELLID.DAT.

3.2.4 FUTURE.DAT

FUTURE.DAT contains future production data that was extracted from the database. The data is in ASCII fixed field format and in sequential order by well identity number. The following table describes the information in each line of the file.

<table>
<thead>
<tr>
<th>Data Description</th>
<th>Field Type</th>
<th>Start Position</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production Date</td>
<td>Character</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Well Identity Number</td>
<td>Integer</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Monthly Gas Production</td>
<td>Integer</td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>Monthly Oil Production</td>
<td>Integer</td>
<td>29</td>
<td>10</td>
</tr>
<tr>
<td>Monthly Water Prod.</td>
<td>Integer</td>
<td>39</td>
<td>10</td>
</tr>
<tr>
<td>Flowing Tubing Pressure</td>
<td>Integer</td>
<td>49</td>
<td>10</td>
</tr>
</tbody>
</table>
3.2.5 NEWFUT.DAT

NEWFUT.DAT is output from DCA1.EXE. It contains new future production data. The data is in sequential order by well identity number and in ASCII delimited format. The delimiter is the space. Each line of data contains future monthly production date, well identity number, gas production, oil production, water production, and flowing tubing pressure. The date is in the form of MM/DD/YY.

3.2.6 RBSFUT.DAT

RBSFUT.DAT is output from FIXFUT.EXE. RBSFUT.DAT contains updated future production data from NEWFUT.DAT and data from wells in FUTURE.DAT that were not processed in the last execution of DCA1.EXE. The data is in sequential order by well identity number and in ASCII delimited format. The delimiter is a comma. Each line of data contains well identity number, future monthly production date, gas production, oil production, water production, and flowing tubing pressure.
3.3 TURBO C EXECUTABLE PROGRAMS

There are five Turbo C executable programs in the project. This section describes the contents of the source files for those programs.

3.3.1 DCASTART.EXE

DCASTART.EXE displays the project start up message.

3.3.1.1 FUNCTION DESCRIPTIONS

main():
Main() in DCASTART.EXE calls all functions necessary to display the project start up message. First, main() calls init_graphics() to initialize the graphics system. Then main() calls demo_border() to draw a border around the screen. Displaymes() is called to display the start up message. Then the program pauses for five seconds. The screen is cleared, the graphics system is closed, and the program terminates.

3.3.1.2 SOURCE FILE

DCASTART.EXE is compiled from the file DCASTART.C. Functions in DCASTART.C are as follows:

int main();
void init_graphics(int *yres);
void errormess4(void);
void errormess5(void);
void displaymes(int screeny);
void demo_border(int yr);

3.3.2 GETDATA1.EXE

GETDATA1.EXE is the program that performs preprocessing on the historical production data extracted from the database.

3.3.1.1 FUNCTION DESCRIPTIONS

main():
Main() in GETDATA1.EXE opens the files RBXTRACT.DAT, DECLINE.DAT, and WELLID.DAT. Then main() calls readdrbdata() to process the information in RBXTRACT.DAT. When program control returns to main(), the data files are closed and the program terminates.
readrbdata():
Readrbdata() reads well data from RBXTRACT.DAT and writes this information with redundant data deleted to the file DECLINE.DAT. The following information is also written to DECLINE.DAT: the major product to water ratio, the minor to major product ratio, monthly cumulative production, total cumulative production, maximum monthly gas production, maximum monthly oil production, maximum monthly water production, maximum water ratio, maximum choke size, maximum flowing tubing pressure, maximum major to minor product ratio, and total number of months of production. Well identity numbers are written to the file WELLID.DAT.

3.3.1.2 SOURCE FILE

GETDATA1.EXE is compiled from the file GETDATA1.C. Functions in GETDATA1.C are as follows:

main();
FILE *openfiler(char *filetoopen);
FILE *openfilew(char *filetowrite);
void errormessl(void);
void readrbdata(FILE *rbdata, FILE *decline,
FILE *wellid);

3.3.3 DCA1.EXE

DCA1.EXE is compiled from the project file DCA1.PRJ. This section describes the data structures, functions, and source files used in DCA1.PRJ.

3.3.3.1 DATA STRUCTURES

There are seven major data structures used by DCA1.PRJ. The data structures are defined in the file DCASTRCT.H. The data structures are described in this section.

3.3.3.1.1 Videostruct

A videostruct structure holds information about various screen parameters. A description of struct videostruct is shown below.

typedef struct
{
  int grphbot;     /*bottom y coordinate on graph grid*/
  int grphtop;     /*top y coordinate on graph grid*/
  int grphtype;    /*flag to indicate whether to plot
                 monthly production versus time (0) or
                 versus cumulative production (1) */
...
int futimlin;  /* x coordinate where future production begins when plotting versus time*/
int futcumlin; /* x coordinate where future production begins when plotting versus cumulative production*/

double xcumratio; /*the number of pixels along the x-axis per unit of scale for cumulative production*/

double cumgrid; /* x-axis scale size when plotting monthly production versus cumulative production*/

double yaxisratio; /*the number of pixels along y-axis per unit of scale for monthly production values*/

} videostruct;

3.3.3.1.2 Futfilestruct

A futfilestruct structure holds information and data from the data file FUTURE.DAT. A description of struct futfilestruct is shown below.

typedef struct
{
    int futeof;  /*flag to indicate EOF for FUTURE.DAT*/
    int futid;  /*well id of future data*/
    int futmonth; /*month of future data*/
    int futyear; /*year of future data*/
    char day[5]; /*day in month of future data*/
} futfilestruct;

3.3.3.1.3 Scalesz

A scalesz structure holds information about y-axis scale size parameters for a data item. A description of struct scalesz is shown below.

typedef struct
{
    double max;  /*maximum scale value for product*/
    double min;  /*minimum scale value for product*/
    double lnmin;  /*logarithm of minimum scale value*/
} scalesz;
3.3.3.1.4 Prodstruct

A prodstruct structure holds miscellaneous information about a data item. A description of struct prodstruct is shown below.

typedef struct
{
    int pf;         /* flag to indicate if production exists*/
    int fi;         /* index into array of future production
        for this product*/
    int type;       /* production type; gas=0; oil=1; water=2;
        ftp=3; ogr=4; choke=5; cum=6*/
    int lc;         /* line color for plotting data item*/
    double pcum;    /* cumulative historical production*/
    double fcum;    /* cumulative future production*/
    scalez sz;      /* structure to hold scale size values for
        data item*/
} prodstruct;

3.3.3.1.5 Identstruct

An identstruct structure holds well identity information and miscellaneous information about each data item. A description of struct identstruct is shown below.

typedef struct
{
    int wellid;      /* well identity number*/
    int majprod;     /* well type; gas = 0; oil = 1*/
    char wellname[36]; /* well name*/
    char wellnum[11]; /* well number*/
    char field[21];  /* field well produces in*/
    char reservoir[21]; /* reservoir within field*/
    char perfs[14];  /* perforations within well bore*/
    char opname[26]; /* operator name*/
    int monthnum;   /* number of months of production*/
    int startyear;  /* year production started*/
    int endyear;    /* year production ended*/
    int endmonth;   /* month production ended*/
    int chkflag;    /* flag for existence of choke data*/
    prodstruct gas; /* miscellaneous information on gas
        production*/
    prodstruct oil; /* miscellaneous information on oil
        production*/
    prodstruct wat; /* miscellaneous information on water
        ratio*/
    prodstruct ftp; /* miscellaneous information on ftp*/
    prodstruct ogr; /* miscellaneous information on minor
        to major product ratio*/
} identstruct;
3.3.3.1.6 Wellstruct

An wellstruct structure holds historical monthly production data. A description of struct wellstruct is shown below.

typedef struct
{
    int mon[MAXMONTH];        /*month in date of production*/
    int yr[MAXMONTH];         /*year in date of production*/
    int xtimplt[MAXMONTH];    /*x coordinate for plotting monthly production versus time*/
    int xcumplt[MAXMONTH];    /*x coordinate for plotting monthly production versus cumulative production*/
    int gasplot[MAXMONTH];    /*y coordinate for monthly gas production*/
    int oilplot[MAXMONTH];     /*y coordinate for monthly oil production*/
    int watplot[MAXMONTH];     /*y coordinate for monthly water ratio*/
    int ftpplot[MAXMONTH];     /*y coordinate for monthly flowing tubing pressure*/
    int ogrplot[MAXMONTH];     /*y coordinate for monthly major to minor product ratio*/
    int chkplot[MAXMONTH];     /*y coordinate for monthly choke size*/

    float cum[MAXMONTH];      /*monthly cumulative production for major product*/
    float gas[MAXMONTH];      /*monthly gas production*/
    float oil[MAXMONTH];      /*monthly oil production*/
    float wtp[MAXMONTH];      /*monthly water production*/
    float wat[MAXMONTH];      /*monthly water ratio*/
    float ftp[MAXMONTH];      /*monthly flowing tubing pressure*/
    float chk[MAXMONTH];      /*monthly choke size*/
    float ogr[MAXMONTH];      /*monthly major to minor product ratio*/
}
wellstruct;

3.3.3.1.7 Futurestruct

An futurestruct structure holds future monthly production data. A description of struct futurestruct is shown below.

typedef struct
{
    int futmon[MAXFUTURE];    /*month in date of future production*/
    int futyr[MAXFUTURE];     /*year in date of future production*/
int futimplt[MAXFUTURE];  /* x coordinate when plotting
   versus time for future data*/
int futcumplt[MAXFUTURE];  /* x coordinate when plotting
   versus cumulative production for future data*/
int futgasplt[MAXFUTURE];  /* y coordinate for future gas
   production*/
int futoilplt[MAXFUTURE];  /* y coordinate for future oil
   production*/
int futwatplt[MAXFUTURE];  /* y coordinate for future water
   ratio*/
int futftpplt[MAXFUTURE];  /* y coordinate for future
   flowing tubing pressure*/
int futogrplt[MAXFUTURE];  /* y coordinate for future minor to
   major product ratio*/
float futcum[MAXFUTURE];   /* future cumulative production
   of the major product*/
float futgas[MAXFUTURE];   /* future gas production*/
float futoil[MAXFUTURE];   /* future oil production*/
float futwtp[MAXFUTURE];   /* future water production*/
float futwat[MAXFUTURE];   /* future water ratio*/
float futftp[MAXFUTURE];   /* future flowing tubing pressure*/
float futogr[MAXFUTURE];   /* future minor to major product
   ratio*/

} futurestruct;

3.3.3.2 FUNCTION DESCRIPTIONS

main():
Main() in DCAMAIN.C opens the file WELLID.DAT and calls the
function getwellid() to get the user's selection of wells to
process. The user's selection of wells is held in the array
wellarray. The variable numwells holds the number of wells
the user chose to analyze. If the user chose to process all
wells, then numwells is set to -1. Then main() closes
WELLID.DAT. The graphics system is initialized.
DECLINE.DAT and FUTURE.DAT are opened for reading.
NEWFUT.DAT is opened for writing. If the user chose to
process all wells, then all wells are processed by a call to
the function processallwells(), otherwise a call is made to
the function processfewwells(). Processfewwells() and
processallwells() will read the data for each well, call the
functions that display the plot of the decline curve on the
screen, and branch to the menu system of the program. After
all wells have been processed, program control returns to
main(). Then main() calls exitproj() to terminate the
program.

init_graphics():
Init_graphics() initializes the graphics system, checks that
a proper monitor is being used, and sets the value of the
variables: v->grphbot, v->grphtop, v->vaxisratio, and v->grphtype. Then init_graphics() calls the function grphscrn().

graphscrn():
Graphscrn() divides the screen into two areas. One area is for text output, and the other area is for graphics output.

exitproj():
Exitproj() exits the project. Exitproj() closes any open files, clears the text portion of the screen, clears the graphics portion of the screen, shuts down the graphics system, and terminates the program.

getwellid():
The purpose of getwellid() is to give the user the option of analyzing all wells or analyzing a few wells in the database. If the user chooses to analyze a few wells, then the identity numbers for those wells are obtained through a call to the function enterwellid(). Duplicate well identity numbers are discarded by a call to the function checkdups().

loadvalid():
Loadvalid() loads well identity numbers from the file WELLID.DAT into an array and counts the number of well identity numbers.

entwellid():
Enterwellid() obtains well identity numbers from the user for wells he chooses to analyze. The function calls checkwellid() to verify the validity of well identity numbers.

checkwellid():
Checkwellid() verifies the validity of a user entered well identity number by ensuring that it is in the array of valid well identity numbers. The array is searched using a binary search algorithm [Ellzey, p.108].

checkdups():
Checkdups() sorts the array of user entered well identity numbers, searches the array for duplicate numbers, and discards duplicate numbers. Checkdups() uses the Turbo C qsort() function which sorts the array using the quick sort algorithm.

processfewwells():
Processfewwells() is a looping routine that reads in well data for each well whose well identity number is in the array wellarray. After data for one well is read, processfewwells() branches to the function prepgraph(). When program control returns from prepgraph(), processfewwells()
will read data for the next well in wellarray. After all wells have been processed, processfewwells() terminates. Program control returns to the function main().

processallwells():
Processallwells() is a looping routine that reads in well data for each well in the database. After data for one well is read, processallwells() branches to the function prepgraph(). When program control returns from prepgraph(), processallwells() reads in data for the next well. After all wells have been processed, processallwells() terminates. Program control returns to the function main().

setparam():
Setparam() sets line colors and data types for each data item.

getidentinfo():
Getidentinfo() reads well identity information such as well name, well number, field, reservoir, etc. from the file DECLINE.DAT into the variable idenstruct info.

prepgraph():
Prepgraph() calls all functions necessary to process well data so that all information is available to produce the decline curve on the screen for a well. Getsuminfo() is called to obtain well summary information. Getfut() is called to obtain future data. Cumulative production for all products is obtained by calling the function getallcums(). Scale sizes along the y axis for data items are determined through the functions calcgrid() and calcratgrid(). Cumulative scale size along the x axis is calculated through the function calcumgrid(). X and y coordinates for monthly production values are determined by calling the function trnsdatalog(). Either grphvstime() or grphvsccum() is called to display the decline curve on the screen. Then prepgraph() branches to the function startopt() which begins the menu facilities of the program. When program control returns to prepgraph(), new future production data is saved through the function savfutdat().

getsuminfo():
Getsuminfo() reads the month and year production started and the maximum monthly production values for the different data items into the appropriate variables. Also, cumulative production from the major product is calculated, and the flag for choke data is set. This information is in the last array entry for each data item in the variable that holds historical production (wellstruct well). Also, the year and month production ended is obtained.
initfut():
    Initfut() initializes the variable that will hold future
data, futurestruct r. All production values are set to
zero. The values of all future months and years are deter-
mined based on the date of last production.

getfut():
    Getfut() reads in future data from FUTURE.DAT for the well
currently being processed. Cumulative production, the major
product to water ratio, and the minor product to major
product ratio are calculated.

getallcums():
    Getallcums() calls the function calccum() to obtain cumula-
tive production for historical and future production for
each data item.

calccum():
    Calccum() calculates cumulative production for a data item.

setflags():
    Setflags() sets the production flag and the future index for
a data item. If production exists for a data item, then the
production flag is set to the number of months of produc-
tion. Otherwise, the production flag is set to zero. If
future cumulative production is zero for a data item, then
its future index is set to zero. Otherwise, it is left
untouched.

calcgrid():
    Calcgrid() calculates the y axis grid size for a data item.
The maximum grid size is not allowed to be less than 1000.
Also, the minimum grid size and the logarithm of the minimum
grid size are calculated.

calcratgrid():
    Calcratgrid() calculates the y axis grid size for a data
item that is a ratio. The maximum grid size is not allowed
to be less than 0.00001. Also, the minimum grid size and
the logarithm of the minimum grid size are calculated.

calccumgrid():
    Calccumgrid() calculates the grid size along the x axis for
cumulative production.

trnsdatalog():
    Trnsdatalog() calls all functions and performs all calcula-
tions necessary so that x and y coordinates for historical
and future production are obtained whether plotting versus
time or cumulative production.
logplot1():
Logplot1() obtains y coordinates for historical monthly production values of a data item.

logplot2():
Logplot2() obtains y coordinates for future monthly production values of a data item.

yaxisrate():
Yaxisrate() calculates the y coordinate of a monthly production value.

xaxiscum():
Xaxiscum() calculates the x coordinate of monthly cumulative production values.

savfutdat():
Savfutdat() saves new future data in the file NEWFUT.DAT.

graphvstime():
Graphvstime() displays the year at the bottom of the graph grid and calls all functions necessary to display the plot of monthly production versus time. Functions called by graphvstime() are drawvchkgrd() or drawechkgd(), gridvga(), prodvxs(), futvxs(), plotchoke(), vgalabels() or egalabels(), and disresults().

graphvscum():
Graphvscum() displays the cumulative production scale at the bottom of the graph grid and calls all functions necessary to display the plot of monthly production versus cumulative production. Functions called by graphvstime() are drawvchkgrd() or drawechkgd(), gridvga(), prodvxs(), futvxs(), plotchoke(), vgalabels() or egalabels(), and disresults().

futtime():
Futtime() plots future monthly production versus time.

futcum():
Futcum() plots future monthly production versus cumulative production of the major product.

drawvchkgrd():
Drawvchkgrd() draws the choke grid and scale for a VGA monitor.

drawechkgd():
Drawechkgd() draws the choke grid and scale for an EGA monitor.
gridvga():
    Gridvga() draws a three cycle semilogarithmic grid on either a VGA or EGA monitor.

prodvxs():
    Prodvsx() plots monthly production versus time or cumulative production if monthly production exists.

futvxs():
    Futvsx() plots future production versus time or cumulative production if future production exists and if cumulative production exists for the major product.

futline():
    Futline() draws a vertical yellow line at the point on the x axis where historical production ends.

plotchoke():
    Plotchoke() plots the choke size.

vgalabels():
    Vgalabels() displays well identity information and the scales along the y axis for all data items to be plotted on a VGA monitor.

egalabels():
    Egalabels() displays well identity information and the scales along the y axis for all data items to be plotted on an EGA monitor.

disresults():
    Disresults() displays cumulative historical and future production below well identity information in the top right corner of the grid.

startopt():
    Startopt() obtains the user's choice when using the Main Menu and processes this choice.

graphsetopt():
    Graphsetopt() gets the user's choice when using the Graph Settings Menu and processes this choice. These options take the user to menus that allow the user to edit the display of the graph on the screen.

analyzeopt():
    Analyzeopt() obtains the user's choice when using the Analysis Menu and processes this choice. These options take the user to menus that allow the user to analyze and extrapolate the decline curve.
graphopt():
Graphopt() obtains the user's choice when using the Graph Type Menu and processes this choice. This menu's options allow the user to change the type of graph displayed.

displayopt():
Displayopt() obtains the user's choice when using the Display Menu and processes this choice. This menu's options allow the user to toggle the display of a data item on and off.

scaleopt():
Scaleopt() obtains the user's choice when using the Rate Scale Menu and processes this choice. This menu's options allow the user to change the scale along the y-axis for a data item. First, the user chooses a data item to process. Then the user chooses a new scale size for the data item. The function changscl() is called to recalculate the scale. Then the graph is refreshed on the screen by calling either grphvstime() or grphvscum().

getnewscale():
Getnewscale() obtains the new scale size from the user for the y-axis for gas production, oil production, and flowing tubing pressure.

changscl():
Changscl() calculates new y coordinate positions of historical and future monthly production values for a data item when the y-axis scale has been changed.

newscaleogr():
Newscaleogr() obtains the new scale size from the user for the y-axis for a data item that is a ratio.

cumopt():
Cumopt() implements the option of changing the scale along the x-axis when plotting monthly production versus cumulative production of the major product. First, cumopt() obtains from the user the value of the new scale size. Then cumopt() calculates the new x coordinate positions of cumulative production for historical and future monthly production values.

declineopt():
The purpose of declineopt() is to get the user's choice when using the Decline Menu and process this choice. This menu's options allow the user to project future production using constant rate, exponential decline, hyperbolic decline, or harmonic decline equations. First, the user chooses a data item to process. Then the user chooses a decline type. The last date of future production is determined for the data
item by a call to the function getlast(). An appropriate function is called for each decline type. These functions prompt the user to enter parameters for the equations and calculate future production for the data item. When program control returns to declineopt(), cumulative production is calculated for the data item by the functions calcsum() and fillcum(). When the chosen data item is gas production, oil production, the major to minor product ratio, or the major product to water ratio, the functions calcratio(), calcminprd(), and calcwatprd() are called which perform calculations to keep data consistent. Then future production is displayed on the screen by either the function futtime() or the function futcum().

choosedecline():
Choosedecline() obtains from the user the type of decline curve equation to use to project production for a data item.

getlast():
Getlast() determines the month and year in the last date of future production for a data item. If there is no future production for a data item, then the last date is the date when historical production ended.

eraseopt():
Eraseopt() implements the "Erase Future Projection" option of the Analysis Menu. First, the user is prompted to choose a data item to process. Then the function erasfut() is called which will interact with the user and perform calculations to erase future production. When program control returns to eraseopt(), cumulative production is recalculated by the functions calcsum() and fillcum(). Then the graph is refreshed on the screen by either the function grphvstime() or grphvscum().

curvefitopt():
Curvefitopt() implements the "Curve Fit Calculations" option of the Analysis Menu. First, the user is prompted to choose a data item to process. If future production exists for the data item, the curve fit will not be performed. The function startfit() is invoked which calls appropriate routines to perform the curve fit and interact with the user. When program control returns to curvefitopt(), cumulative production for the data item is calculated by the functions calcsum() and fillcum(). When the chosen data item is gas production, oil production, the major to minor product ratio, or the major product to water ratio, the functions calcratio(), calcminprd(), and calcwatprd() are called which perform calculations to keep data consistent. Then either futtime() or futcum() is called to display future production on the screen.
fillcum():
Fillcum() calculates future cumulative production for the major product. Also, fillcum() determines the x axis coordinate position for future monthly cumulative production values by calling the function xaxiscum().

getchange():
Getchange() gets the user decision as to whether to change the future projection for the minor product or ratio when a change has been made to future production of the major product.

calcratio():
Calcratio() calculates the future projection of the minor to major product ratio using the future projections of the major and minor products. Also, the y coordinate position is determined for the future monthly values of the ratio by calling the function yaxisrate().

calcminprod():
Calcminprod() calculates the future projection of the minor product using the future projections of the major product and the minor to major product ratio. Also, the y coordinate position is determined for the future monthly values of the minor product by calling the function yaxisrate().

calcwtrpd():
Calcwtrpd() calculates future production for water using the future projections of the major product and the major product to water ratio. Also, cumulative water production is calculated by calling the function calcsum().

erasfut():
Erasfut() erases future production for a data item. Erasfut() obtains the month and year when future production is to be erased by calling the function getyrmon(). Then the number of gap months between the last historical production date and the erasure date are obtained by calling the function calcmnth(). Then all future monthly production values and y coordinate positions are set to zero starting at the month when the erasure date begins.

constant():
Constant() calls all functions necessary in order to project production at a constant rate for a data item. Constant() obtains the unit of production, the rate of constant production, and the dates constant production is to begin and end. From this information, the number of months before constant production begins (gapmonths) and the number of months of constant production (constmonths) are determined. Constant sends this information to the function constfut() which
actually sets the future monthly production values to the constant rate.

constfut():
Constfut() sets the value of future monthly production rates to a constant rate for a data item. Also, the y coordinate position of the future monthly production values is determined. The future production index is updated.

exponential():
Exponential() calls all functions necessary in order to project production using exponential decline curve equations. Exponential() obtains the date exponential decline is to begin. The number of gap months between the last date of historical production or of future production and the beginning date are determined. Then exponential() obtains the unit of production, the initial production rate, the decline rate unit, the effective decline rate, and an end condition. If the end condition is a production rate, exponential() obtains the final production rate. If the end condition is a date, exponential() obtains the ending date. In either situation, the number of months of exponential decline is determined. Then exponential() calls the function expfut() which actually calculates future monthly production values for a data item.

calcexpmn():
Calcexpmn() calculates the number of months of exponential decline using equation 5 in Section 2.2.3.15.

expfut():
Expfut() calculates future monthly production values for a data item using equation 4 in Section 2.2.3.15. Also, the y-axis coordinate for future monthly production values is determined.

hyperbolic():
Hyperbolic() calls all functions necessary in order to project production using hyperbolic decline curve equations. Hyperbolic() obtains the date hyperbolic decline is to begin. The number of gap months between the last date of historical production or of future production and the beginning date are determined. Then hyperbolic() obtains the exponent to use in hyperbolic equations. Also, hyperbolic() obtains the unit of production, the initial production rate, the decline rate unit, the initial effective decline rate, and an end condition. If the end condition is a production rate, hyperbolic() obtains the final production rate. If the end condition is a date, hyperbolic() obtains the ending date. In either situation, the number of months of hyperbolic decline is determined. Then hyperbolic() calls the
function hypfut() which actually calculates future monthly production values for a data item.

gethypcst():
Gethypcst() obtains from the user the exponent to use in hyperbolic decline curve equations. The exponent must be in the following range:
0 < exponent < 10, exponent ≠ 1

calchypmn():
Calchypmn() calculates the number of months of hyperbolic decline using equation 8 in Section 2.2.3.15.

hypfut():
Hypfut() calculates future monthly production values for a data item using equation 7 in Section 2.2.3.15. Also, the y axis coordinate for future monthly production values is determined.

harmonic():
Harmonic() calls all functions necessary in order to project production using harmonic decline curve equations. Harmonic() obtains the date harmonic decline is to begin. The number of gap months between the last date of historical production or of future production and the beginning date are determined. Harmonic() obtains the unit of production, the initial production rate, the decline rate unit, the initial effective decline rate, and an end condition. If the end condition is a production rate, harmonic() obtains the final production rate. If the end condition is a date, harmonic() obtains the ending date. In either situation, the number of months of harmonic decline is determined. Then harmonic() calls the function harfut() which actually calculates future monthly production values for a data item.

calcharmn():
Calcharmn() calculates the number of months of harmonic decline using equation 14 in Section 2.2.3.15.

harfut():
Harfut() calculates future monthly production values for a data item using equation 13 in Section 2.2.3.15. Also, the y axis coordinate for future monthly production values is determined.

getryrmon():
Getryrmon() obtains the year and month in a date such as the date future production is to begin, the date future production is to end, or the date when future production is to be erased. The date future production is to begin must be a date later in time than the last date of historical production or of the date of last future production. Then date
future production is to end must be a date later in time than the date production began for the period of decline. The date when future production is to be erased must be a date later in time than the last date of historical production.

calcmnths():
Calcmnths() calculates the number of months between two dates. Each date is represented by a month and a year.

chooseunit():
Chooseunit() obtains the unit for the production rate from the user. The user has several choices for oil and gas production. The user must use default units for the major to minor product ratio, the major product to water ratio, and flowing tubing pressure.

getrate():
Getrate() obtains production rates from the user. All rates are converted to monthly values. The converted rate must be greater than zero.

choosedeclunit():
Choosedeclunit() obtains the unit of the effective decline rate from the user.

getdecline():
Getdecline() obtains the effective decline rate from the user as a decimal number. The decline rate must be greater than zero and less than one. Then the effective decline rate is converted to the nominal decline rate. Equation 1 in Section 2.2.3.15 is used for exponential decline. Equation 6 is Section 2.2.3.15 is used for hyperbolic decline. Equation 12 in Section 2.2.3.15 is used for harmonic decline. Once the nominal decline rate is obtained, the rate is converted to a monthly value based on the following equations:
\[
\text{annual nominal decline rate} / 12 = \text{monthly nominal decline rate} \\
\]  
[Nind, p. 35]

daily nominal decline rate * 365 = \text{monthly nominal decline rate} \\
[derived from the above equation]

getendcond():
Getendcond() obtains from the user the type of condition used to end a decline period. The end condition can either be a production rate or a date.