Enhancement and Support of Computer Aids for Fisheries Management

Final Technical Report

MRAG Ltd

March, 2001

Final Report - Administrative Details

DATE COMPLETED : 31 / 3 / 01

TITLE OF PROJECT:	Enhancement and support of computer aids for fisheries management
PROGRAMME MANAGER (INSTITUTION):	MRAG Ltd 47 Prince's Gate London SW7 2QA
START DATE	1/11/98
FINISH DATE	30/10/00
AUTHOR	Dr G. Kirkwood

Acknowledgements

Richard Aukland undertook virtually all the design and programming work required to produce the new versions of CEDA and LFDA in Visual Basic. Graham Pilling and Ashley Halls assisted in the testing of the two software packages. Steve Zara's help was also invaluable in making the necessary final changes before release of the software packages.

Table of Contents

Final Technical Report

1. Executive Summary

CEDA (Catch and Effort Data Analysis) and LFDA (Length Frequency Distribution Analysis), are two pc-based software packages, designed for use by fishery officers in developing countries when carrying out fishery stock assessments, that were developed under the FMSP and first released in 1992. Both were developed for the then universally used MS-DOS operating system. More than 50 copies of the software packages were distributed to scientists and scientific organisations in developing countries during the year after their first release. Feedback from users of the original versions of the packages and experience gained during a follow-up adaptive project subsequently led to revisions of the software packages being made. Revised versions (CEDA Version 2.01 and LFDA version 4.01), which still used MS-DOS, but incorporated a Windows-like menu system, were released in 1995. By 1999, more than 150 copies of the revised packages were in use in developing countries worldwide. More recent feedback has been that the packages are getting harder to use with the modern Windows 95 or later operating systems, and the lack of a Windows-standard user-interface. Accordingly this project aimed to produce revised Windows versions of the two packages, which would then be able to be distributed to users via the Internet via the MRAG/FMSP website.

Rewritten in Microsoft Visual Basic, the two software packages have standard Windows user interfaces and a very extensive context-sensitive on-line help systems. They also incorporate comprehensive tutorials illustrating the use of the packages, and separate guides on statistical issues and on use of population dynamics models. The LFDA packages allows users to estimate non-seasonal and seasonal growth curves from length frequency data using three alternative methods. Given these estimated growth curves, further analysis allows estimation of total mortality rates and estimation of age frequency distributions. The CEDA package allows users to fit a number of alternative population dynamics models to catch and effort data, thereby estimating stock sizes and important management quantities such as the maximum sustainable yield.

The improved scientific advice available through the use of these packages will considerably enhance the likelihood of sustainable management of vital fishery resources, which in developing countries often represent major sources of animal protein, employment and income.

2. Background

FMSP Projects R.4517 and R.5050CB resulted in the production of two pc-based software packages, designed for use by fishery officers in developing countries when carrying out fishery stock assessments. These packages were first released in 1992. Both were developed for the then universally used MS-DOS operating system.

The two packages, which do not require the user to have programming skills, were:

CEDA - Version 1.0. Catch and Effort Data Analysis: A software package for analysing catch, effort and abundance index data. The outputs included estimates of the maximum sustainable yield and replacement yield, current and unexploited biomasses, and other key population dynamics parameters.

LFDA - Version 3.0 Length Frequency Distribution Analysis: A software package for analysing length frequency data. The outputs included estimates of growth parameters and mortality rates.

As a result of project R.4517, more than 50 copies of the software packages were distributed to scientists and scientific organisations in developing countries during the year after their first release. Feedback from users of the original versions of the packages and experience gained during the adaptive project R.5050CB subsequently led to revisions of the software packages being made. Revised versions (CEDA Version 2.01 and LFDA version 4.01), which still used MS-DOS, but incorporated a Windows-like menu system, were released in 1995.

Takeup of the CEDA and LFDA packages has been reported annually in the FMSP annual report from 1993 to 1999. As shown in the tables below, the takeup has grown steadily over that period, and more than 150 copies of the revised packages are now in use in developing countries worldwide.

Table 2.1 Regional and Institutional Breakdown of CEDA and LFDA Dissemination in Developing Countries between 1993 and 1998

C. America/Caribbean

Institute Type	93	94	95	96	97	98
University	3	4	4	4	5	5
Fisheries Laboratory	5	7	8	8	15	15
Development Agency	0	1	1	1	2	2
Other	0	0	0	2	7	7
Total	8	12	13	15	29	29

South America

Institute Type	93	94	95	96	97	98
University	0	1	2	2	2	2
Fisheries Laboratory	3	4	6	6	7	7
Development Agency	1	1	1	1	2	2
Other	1	1	1	3	4	4
Total	5	7	10	12	15	15

Africa

Institute Type	93	94	95	96	97	98
University	2	2	3	4	4	4
Fisheries Laboratory	7	9	12	12	12	12
Development Agency	9	11	13	13	13	13
Other	0	0	0	2	2	2
Total	18	22	28	31	31	31

Bangladesh/India

Institute Type	93	94	95	96	97	98
University	3	5	9	10	10	10
Fisheries Laboratory	1	2	4	4	4	4
Development Agency	4	4	4	4	4	4
Other	3	3	3	3	3	3
Total	11	14	20	21	21	21

Indian Ocean

Institute Type	93	94	95	96	97	98
University	0	0	0	1	1	1
Fisheries Laboratory	2	3	4	5	5	6
Development Agency	5	7	8	8	8	8
Other	0	0	0	0	0	0
Total	7	10	12	14	14	15

SE Asia

Institute Type	93	94	95	96	97	98
University	0	2	4	4	5	5
Fisheries Laboratory	4	7	8	8	8	8
Development Agency	5	7	8	9	9	9
Other	0	0	0	0	0	0
Total	9	16	20	21	22	22

Total

Institute Type	93	94	95	96	97	98
University	8	14	22	25	27	27
Fisheries Laboratory	22	32	42	43	51	52
Development Agency	24	31	35	36	38	38
Other	4	4	4	10	16	16
Total	58	81	103	114	132	133

The much larger and still-expanding user community has led to a much greater demand on MRAG for advice and support in the everyday running of the software packages. Until now, these requests have been handled on an *ad hoc* basis, but the continuing high level of demand for support requires that a more formal and efficient mechanism for support be developed. It is also clear that there has been considerable secondary distribution of the software packages by those individuals to whom the software was originally sent by MRAG, but the full extent of this and the needs of these users are not well documented. Part of this project will enable MRAG to enhance their World Wide Web (WWW) site to allow users to download copies of the software and manuals and to provide feedback to MRAG about its usage. This will enable us to build up a database of our users.

In addition to requests for support, MRAG has also received considerable feedback from users of the software on desired amendments and extensions to the software. Much of the more recent feedback has been on the desirability of Microsoft Windows versions of the software. This would enable users to interface CEDA and LFDA more easily with the most common software packages in use such as spread sheets, word processors and databases. In addition, the more modern operating systems used even in developing countries have greater and greater difficulty in running old MS-DOS programs. These needs are addressed by the main purpose of the project, which involves development, testing and dissemination of revised Windows-based versions of the two software packages.

3. Project Purpose

Building on feedback already received from users, develop, test and distribute revised Windows-based versions of the CEDA and LFDA software. Enhance the MRAG web site to allow remote downloading of programs and manuals for CEDA and LFDA, and as a mechanism for users to register their use of the software and to provide feedback to MRAG about the software.

4. Research Activities

The research activities consisted of redesigning the two software packages for a full Windows environment, programming them in object-oriented Microsoft Visual Basic, testing, incorporation of comprehensive on-line context-sensitive help systems and including detailed expository example analyses using the software packages. The software will be disseminated through the newly-developed FMSP web site.

5. Outputs

The output from this project is the completed revised CEDA Version 3.0 and LFDA Version 5.0 software packages, including the online help files and example analyses. A CD containing the software packages is enclosed with this report.

To illustrate the capabilities of the software packages and their use, the example analyses are reproduced in this section.

5.1 LFDA Tutorial

5.1.1 Introduction

The object of this tutorial is to guide you through the analysis of an example set of length frequency data using LFDA version 5. If you are new to LFDA, this will help you become familiar with the procedures necessary to use the package. If you have used previous versions of LFDA, you will notice that there have been a number of changes to the user interface in version 5, so working through the tutorial will introduce you to most of these changes.

We have deliberately chosen to use in the tutorial a simulated length frequency data set, with strong modes and fairly clear progressions between them. This choice was made to give you confidence in using the package and a clear idea of how each of the methods work, but also in order to illustrate that even in the most favourable of circumstances, identifying the 'best' estimates of growth parameters or mortality rates from length frequency data is not straightforward. You will see that the different methods of analysis can give different results, and you will be able to compare them with the ‘true’ values used to generate the simulated data. No matter how good your data are -- and they are unlikely to be as good as the simulated data used here -- a degree of uncertainty will always remain in estimates of growth and mortality parameters based on length frequency data. To obtain reliable estimates of parameters, there must be a pattern in the length frequency data, but this pattern is often not strong or clear. In order to select the most appropriate single set of estimates (or alternative sets of estimates) from the results obtained using the LFDA package, you must use both common sense and judgement, and bring to bear any other relevant biological information you may have about the fish under study. Do not believe a set of results just because they were produced by a computer!

During the course of the tutorial, we will illustrate the use of most of the options available in LFDA, but not all of them. A complete description of all the options available is contained in the Reference section of the Help files.

5.1.2 Loading and inspecting data

There are a number of different ways to load data into LFDA, and these are described in detail in the reference section of this manual. The most common way of loading data is to import it from an ASCII text file. You can create ASCII files using an ordinary text editor or you can export them from a spreadsheet, database or word processor. The data which we are going to use for the tutorial has been saved in the ASCII file TUTOR.TXT. You can view any ASCII file using the a text viewer such as Notepad. If you open the file TUTOR.txt in Notepad, the first few lines should look like the table below:

Simulated Tutorial Dataset

	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
20.0	215.0	0.0	0.0	0.0	0.0	241.0	0.0	0.0	0.0	0.0
25.0	128.0	0.0	0.0	0.0	0.0	113.0	0.0	0.0	0.0	0.0
30.0	71.0	1.0	0.0	0.0	0.0	72.0	3.0	0.0	0.0	0.0
35.0	30.0	9.0	0.0	0.0	0.0	20.0	7.0	0.0	0.0	0.0
40.0	6.0	19.0	1.0	0.0	0.0	7.0	20.0	4.0	0.0	0.0
45.0	4.0	30.0	3.0	4.0	0.0	4.0	39.0	7.0	3.0	0.0
50.0	0.0	56.0	7.0	3.0	0.0	0.0	72.0	6.0	2.0	0.0
55.0 ...										

The first line of such a file contains a description of the dataset, and it will be used by LFDA as a title for this set. The second contains the sample timings, that is, the relative times at which the samples were taken. Often, you will have stored this information in the form of a date (e.g. 24th January 1990). In the LFDA package, we have assumed that the primary unit of time is a year, and the package needs to know how many years or fractions of a year have elapsed between a nominal starting date and the time at which the length frequency distribution was taken. Thus, suppose you started collecting length frequencies in 1990. 1st January 1990 is a sensible date from which to measure, so if your first distribution was taken on 24th January 1990, then the sample timing for that sample is 24/365 = 0.066. If the next distribution was taken on 3rd March, then the sample timing for that distribution would be (31+28+3)/365 = 0.170. The sample timing on 3rd March of the following year would be 1.170. Parameters like t0 are estimated relative to the 0 sample time, i.e.1st January 1990.

The rest of the rows in the file have the same format. The first number in the row is the length class, and the rest of the entries in the row are the numbers of fish in that length class in each of the samples. The result is that the sample length distributions form the columns of the table, with the first column being the length classes. Length class intervals must all be of the same size.

You can import your own data into LFDA provided it is in an ASCII file in the same format as this example file (you can have more columns or rows, though). Word processors and spreadsheets can all save or export data into ASCII format, so if you have entered your data using such a package, look in its documentation to see how to do this.

Now that you have looked at the tutorial data, we will import it into LFDA. Select the file menu File | Open. A dialog box will appear on the screen, listing the names of all the *.lf5 files in the current directory. Use the pull down list to change the shown file types to files of type Text (*.txt). Select TUTOR from the list of text files.

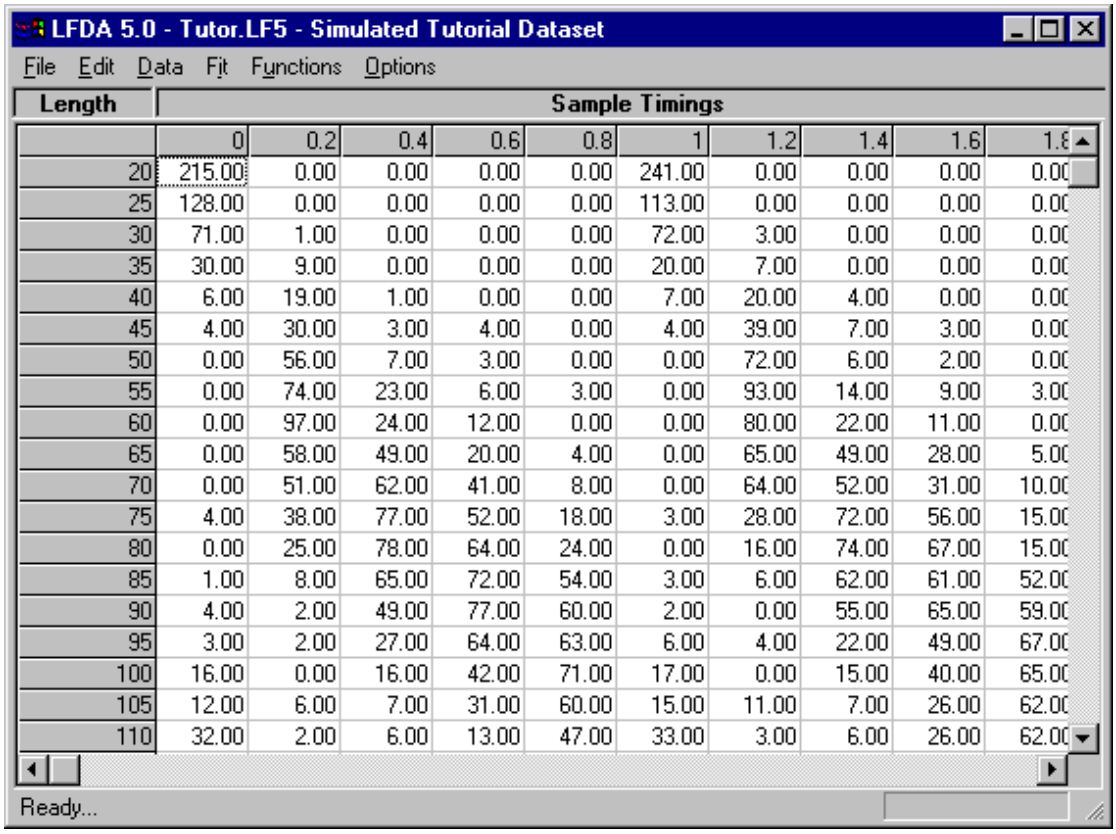


Figure 5.1.1 Main Application window showing imported dataset.

The data set TUTOR will then be imported into LFDA and you will be asked to provide a filename for the newly imported dataset. We suggest that you enter the filename Tutor. The newly imported dataset will then be saved as an LFDA file with the extension *.LF5. The main data sheet window should now appear as shown above in Figure 5.1.1

If you now move to the Data menu, you will see a number of options for viewing the data. These options are discussed in detail in the Reference section. For the moment, we will just plot the data as length-frequency histograms by selecting Data | Plot Data.

The plot should look like Figure 5.1.2 below. Examine these histograms carefully as, in the absence of other information, they will provide the primary clues for a first guess at possible ranges for growth curve parameters that you will need in the next section. In particular, the upper ends of the histograms should give you some idea of possible values for L.

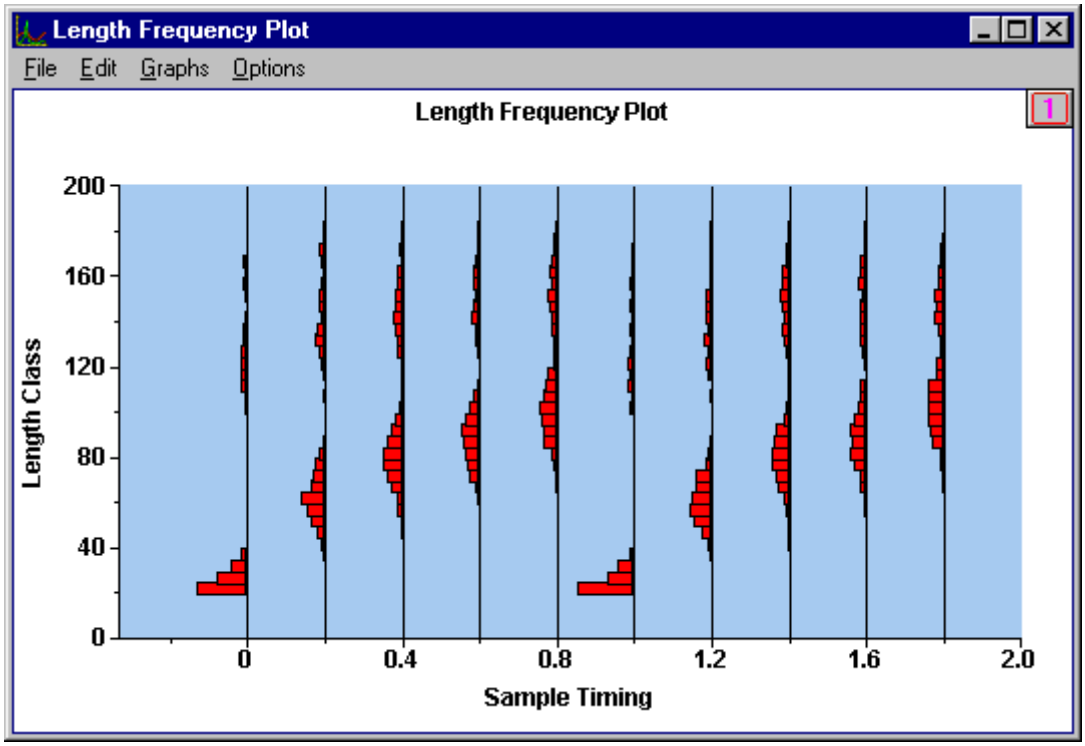


Figure 5.1.2 Sample Length Frequency Histograms

5.1.3 Estimation of non-seasonal growth parameters

You are now ready to estimate some growth parameters. The LFDA package allows you to choose between three different growth functions that you can try to fit to your data: the von Bertalanffy, Hoenig and Pauly functions. The von Bertalanffy model is the simplest, requiring three parameters to be estimated. The latter two models make allowance for seasonal growth and so require two more parameters and therefore better data. Whichever function you choose, there are alternative methods for estimating which parameters will provide the best fit to your data. The three methods in LFDA are: Shepherd's Length Composition Analysis (SLCA); the Projection Matrix Method (PROJMAT); and a version of the Electronic LEngth Frequency ANalysis method (ELEFAN).

We will look at a number of combinations of model and method in the course of this tutorial. This is not only to illustrate the different procedures, but also to demonstrate the way you should work in analysing your own data. Each method has its strengths and weaknesses, and different data sets will respond best to different methods. For this reason it is sensible to try all methods and all suitable models on your data. If you can get agreement between the methods, it will give you that much more confidence in the accuracy of your parameter

estimates. For a detailed discussion of the methods and models please refer to the Help file sections on Estimating Growth Parameters, and the Technical Appendix.

If you have read the chapters on Estimating Growth Parameters, you will find the tutorial much easier to follow. However, for those who are impatient to continue, some of the most important concepts are repeated here. **But you must be very familiar with the contents of that section before you try to use LFDA on your own data.**

A score function in LFDA is something that takes a model (e.g. von Bertalanffy) with its specific parameters (e.g. values for K and $L_{¥}$), then looks at your data and gives you a number which tells you how likely it is that your data comes from a stock with that growth function. You calculate the score function for lots of different parameters, and the higher the score function, the more likely it is that your stock's growth follows that growth model with those parameters. The three methods, SLCA, PROJMAT and ELEFAN, all have different score functions.

The basic idea behind estimating the growth parameters is to find the combination of parameters that maximises a specified score function. Although the score functions for the three methods are different, they all, in some way, measure the goodness of fit between the observed length frequencies and those that would be expected if growth followed the specified model and parameters. However, it is not easy to find that combination of parameters that gives the maximum score function. To get started, we calculate the value of the score function over a specified grid of values of K and $L_{¥}$. Then we can see in which portion of the grid the maximum score function lies, and specify a new, finer grid covering only that area. Once we have narrowed down the possible ranges of K and $L_{¥}$, we can use LFDA's built-in automatic maximisation to determine the values of K and $L_{¥}$ at which the score function is maximised.

5.1.3.1 Shepherd's Length Composition Analysis (SLCA)

Shepherd's method compares each observed length-frequency distribution with a length frequency distribution that would be expected for given values of the von Bertalanffy growth parameters $L_{¥}$ and K. A goodness-of-fit score is then calculated using a certain test function (Shepherd, 1987; see also Technical Appendix). Large, positive score values indicate that the expected length-frequency distribution matches well with the observed data. The best estimates of K and $L_{¥}$ are those that correspond with a maximum value of the score function. SLCA is only applicable to the von Bertalanffy growth model because it does not handle seasonal growth.

As we discussed in the previous section, the first step is to define a grid over which the score function must be calculated. The different estimation methods have different score functions, so at this stage we also have to choose the model and method to use. Select the **Fit | Score Function Grid** menu and you should see the **Grid Evaluation** dialogue box as shown in Figure 5.1.3

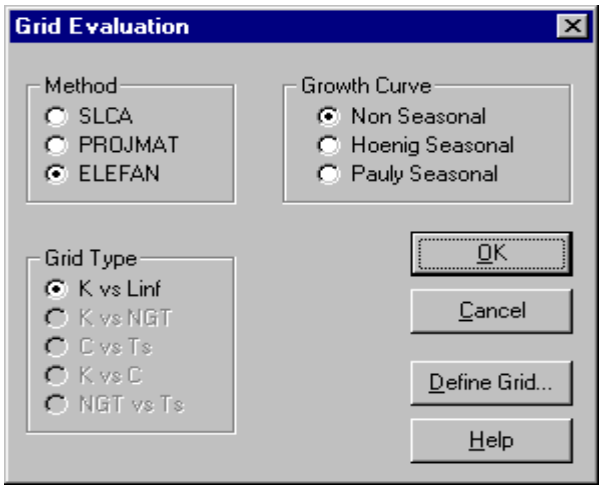


Figure 5.1.3 Grid Evaluation Dialogue Box

In the top left selection of options you can specify whether to use the SLCA, PROJMAT or ELEFAN method. SLCA is probably the one currently selected. If not, click on the circle or use the cursor keys to move the highlight and dot between the three methods. Next we will specify a growth model. The Hoenig and Pauly models are for stocks with seasonal growth which cannot be estimated using SLCA. We therefore can only use the von Bertalanffy model, so select this option as before.

The shape of the von Bertalanffy curve can be completely specified by the two parameters K and L_{∞}, since t_0 can be estimated from them. The Hoenig and Pauly models have four independent parameters to be estimated, so it is not possible to display a grid of all parameters on the screen. If you were using these models you would therefore have to specify which parameters you would like to vary over the grid, while the other parameters would remain fixed at values you had chosen. This is the purpose of the third option group, but since we are using a von Bertalanffy model, K vs L_{∞} is the only appropriate option and should remain selected. When you are satisfied select the **Define Grid** button to display the Grid Options dialog box shown below in Figure 5.1.4.

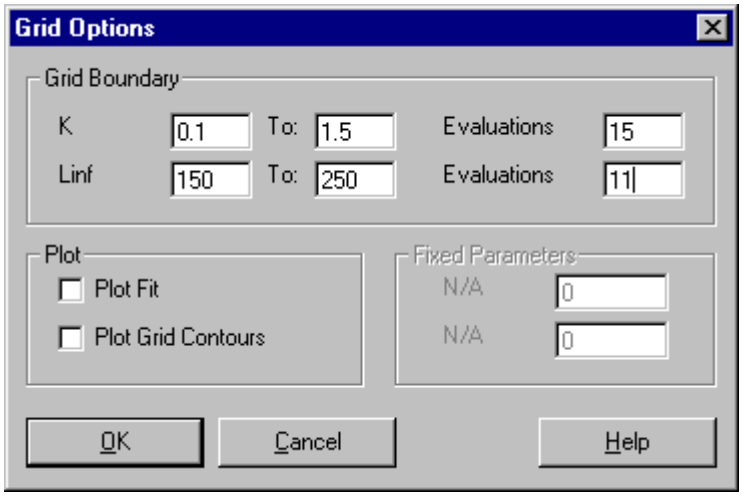


Figure 5.1.4 Grid Options Dialog Box

The Grid Options dialog box asks you to specify the rectangular grid of values by giving the smallest and largest values of K and L_{∞} that you think might possibly be consistent with the data, and the number of values (evaluations) in between that should be calculated. Experience with estimating growth parameters suggests that it is wise at first to choose as big

a grid as possible when starting the estimation process. Recalling the histograms of the length frequencies that we plotted earlier, it would seem reasonable that the lowest possible value for L_∞ is 150. (Remember, the von Bertalanffy parameter L_∞ here measures the **average** maximum length in a population of fish, not the length of the biggest fish of the lot). Picking an upper value is less easy, so let us try 250, which ought to be big enough. Picking a range for *K* is also not easy. A trial of say 0.1 to 1.5 might be a sensible first guess. Note that it is very simple and quick to return to this calculation again and again to hone down your ranges. Finally, you are asked for the number of evaluations for each range of *K* and L_∞. A sensible first number would be 15 for *K* and 11 for L_∞. To be absolutely clear, if you ask for 11 evaluations using L_∞ between 150 and 250, the package will evaluate at L_∞ = 150, 160, 170, 180, 190, 200, 210, 220, 230, 240 and 250. Combined with 15 evaluations for K, that makes 15x11=165 evaluations in all.

So now fill the grid evaluation information into the dialog box in the appropriate spaces. When you have finished the dialog box should look like figure 5.1.4 above. Now click OK or press Enter to accept the grid options. Now, if you press the OK button on the Grid Evaluations dialog box LFDA will immediately start calculating the SLCA score function values for each point on the grid, displaying a progress bar with an estimate of time remaining. Time should not be a problem for SLCA, but it may be for some of the other methods if you have a slower machine.

When the calculations are finished LFDA presents the Parameter Estimates window, which gives the details of, the grid calculated and the best parameter estimates obtained using the grid. The best score is obtained for *K*=0.7 and L_∞ =220, with *t*0=-0.163. Notice the lines on the panel saying 'Fixed Parameters : None' and 'No maximisation performed'. These will change later as we start using other features of LFDA. This window can be retrieved later, if you wish, by selecting **Fit | View Parameter Estimates**.

LFDA can plot your length frequency data together with the von Bertalanffy curve using the parameters estimated at the grid point with the highest score.. In this case, this was for *K*=0.7 and L_∞ =220. If you select the menu item **Plot | Growth Curve** from the Parameter Estimates window you should see a plot like the one shown in Figure 5.1.5 . The title of the plot contains the curve's parameters, including the corresponding t_0, -0.163. (On screen, t_0 is sometimes referred to as tz or Tzero). Eventually, we hope the growth curve will fit the histograms well, passing through the middle of the bases of the modes. We have only just started, however, so we cannot expect too much of the fit at this stage. If you wish, you can redraw this plot later by selecting **Data | Plot Fit** from the main application window menu and accepting the latest 'best' parameters which will have been automatically entered into the dialog box.

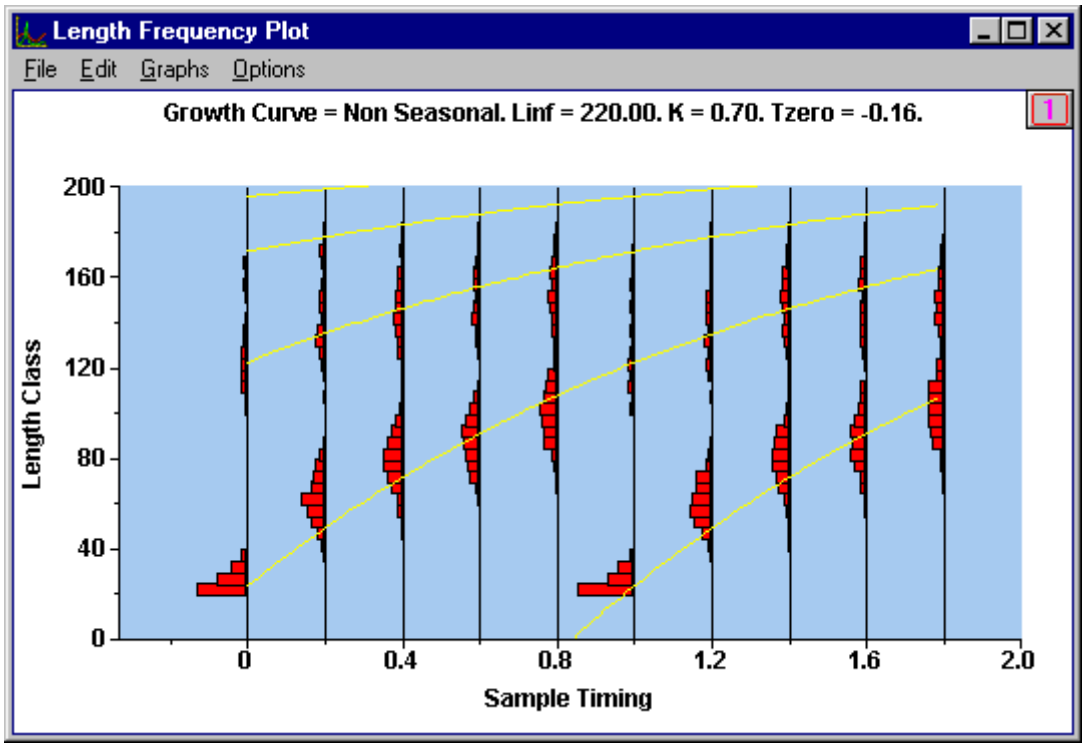


Figure 5.1.5 Plot of Length Frequencies with fitted von Bertalanffy growth curve

To view the grid evaluations calculated by LFDA you should now return to the Parameter Estimates window and select the **Grid Data | View** menu item. A new window with the caption "Grid Search Data" will appear which contains a grid of the values of the score function for each grid point. The maximum score function, 374.982, is highlighted in red, while the minimum is in blue. Check that the maximum is at K=0.7 and L_{∞} =220, and look at the score functions values around this point. Note that these estimates are extremely unlikely to be good, since we were evaluating a fairly coarse grid. We are only seeing the best of the 165 K and L_{∞} combinations considered so far. This first step is just to get a general idea of where to search further.

If you try to scan through this table, you will rapidly discover that it is difficult to take it all in. This is normal. To help interpret the table, LFDA can produce a two dimensional contour plot. This is rather like a weather or a topographic map on which it is easy to identify high and low regions. Return to the Parameter Estimates window and select the menu item **Plot | Grid Data** and a contour plot similar to that shown in Figure 5.1.6 will now appear on the screen.

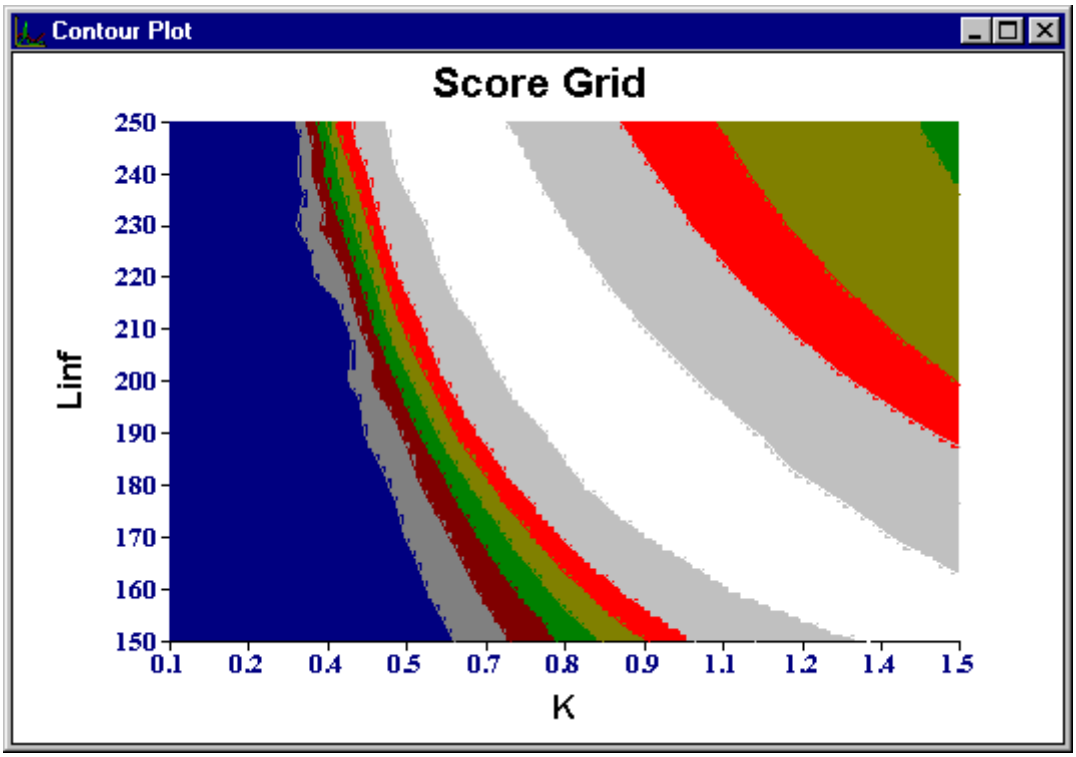


Fig 5.1.6 Contour Plot of Score Functions

On this contour plot, the dark areas represent low values of the score function and the light areas represent high values. So we are seeking to identify values of K and L_{∞} that correspond to the lightest areas. The white region in the contour spreads in a broad curve across the centre of the screen, corresponding, as expected, to a wide (but inversely correlated) range of K and L_{∞}.

The contour plot works by assigning a colour to those grid points with score function values within a given range, and the contours are drawn by interpolating between the calculated grid points. Initially, the score ranges corresponding to each colour are quite wide, but it is also possible to concentrate only on the high score function values and allocate colours to smaller ranges. This gives a more detailed contour plot and is accessed using the right hand mouse button. If you press the right hand mouse button down over the contour plot, you will see that the white area narrows, and recedes from the bottom right hand corner. Pressing the right hand mouse button again narrows the white range still further. You should not do this more than once or twice for this size grid, or you will start seeing detail that is not really there, but is just an artifact caused by the coarseness of the grid calculation that we did. If you repeat this again, the white area will break up into patches about the size of the grid, i.e.10cm in L_{∞} and 0.1 in K. This is completely artificial and does not tell you anything about your data, so you must be careful to watch out for this effect. Close down the contour plot and redraw it at the original level of detail by selecting the menu item **Plot | Grid Data**, and press the right hand mouse button down twice to magnify it again.

We do seem to have been too conservative in the range specified for K. From this contour plot, it would seem that a better range would be 0.5 to 1.2. It also suggests that the estimate L_{∞} will be higher than 180 cm. The white area runs off the screen as if L_{∞} could be higher than 250 cm, but with the length frequency histograms showing a maximum length around 190 cm, it is unlikely that L_{∞} can be so high. (Of course for a heavily exploited stock this would not necessarily be true, so you must use your knowledge of the biology of the stock to make such decisions). We will keep an eye on this, however, so that if we detect maxima near to such large values then we can extend the range of L_{∞}'s in our grid search upwards.

Let us now calculate a new score function grid, using the revised ranges for K and L_{∞}. Firstly close down all windows which remain open except for the main application window. Now select the menu item **Fit | Score Function Grid** to define a new grid. The Grid Evaluation dialog box will reappear specifying the model and method to be used. Make sure that the options selected are still Method = SLCA, Growth Curve = Non Seasonal and Grid Type = K *vs* L_{∞}. Press the Define Grid button to show the Grid Options dialog box. Here you should fill in the new values: K from 0.5 to 1.2 with 15 steps and L_{∞} from 180 to 250 also with 15 steps. Press OK to close this dialog box and then press OK on the Grid Evaluation dialog box to perform the grid search..

As an aside, you may wonder why we are using odd numbers of steps like 11 and 15. There is no real need for it, but it is good practice to have grid evaluations take place on round values of K and L_{∞}. It doesn't matter mathematically, but a grid maximum at a value of L_{∞} like 183.734 can give you a false sense of accuracy. If you wish to divide a range into 10 portions you must evaluate 10+1=11 grid points.

The new grid evaluation still gives the maximum score function at K=0.7 and L_{∞} =220. Select the item menu **Plot | Grid Data** in the Parameter Estimates window to draw the new contour plot shown in Figure 5.1.7.

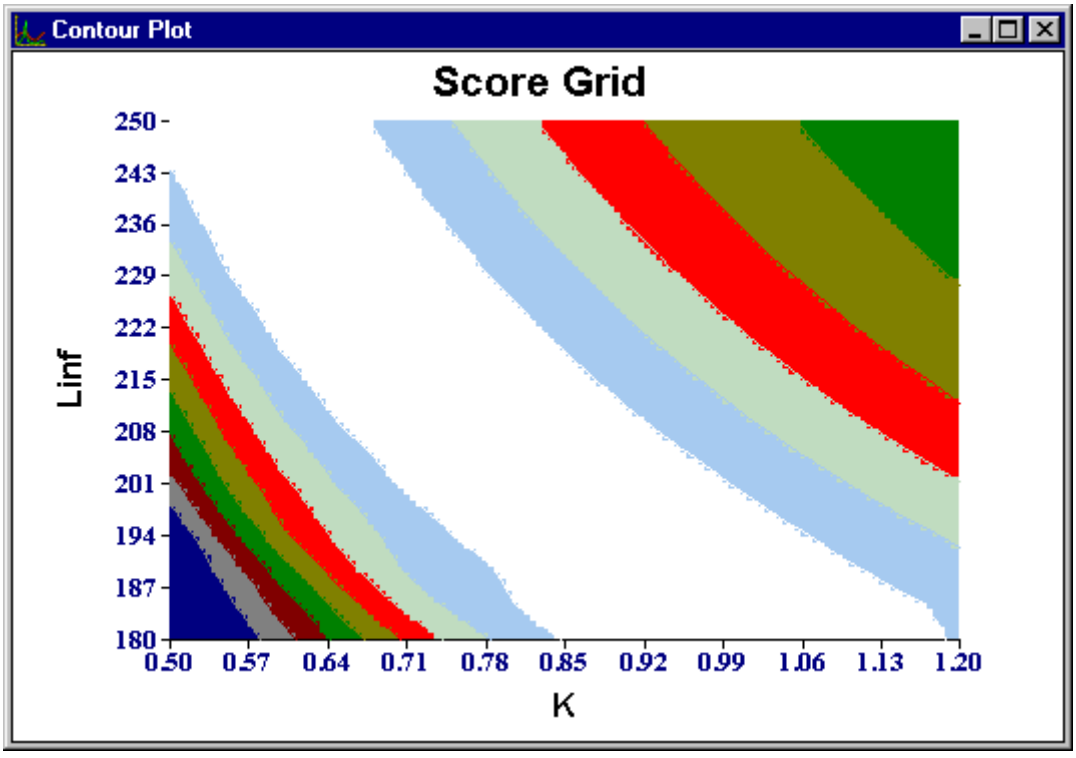


Figure 5.1.7 New Contour Plot Redrawn from New Grid Evaluations

We seem to have picked the right range this time. The plot indicates a band of high score function values running from the bottom right of the plot to the top left. These slightly banana-shaped contours are typical of what you will see for almost every data set, and they reflect the high negative correlation between the estimates of K and L_{∞}. You should always seek to generate contour plots with this shape.

As the contour plot appears now, it seems difficult to narrow down the range of values of K and L_{∞} which may correspond to a maximum. Once again, we need more detail; even more so because we have a narrower range of values around the maximum. Press the right hand

mouse button down over the contour plot three times. A new plot will be shown as in Figure 5.1.8.

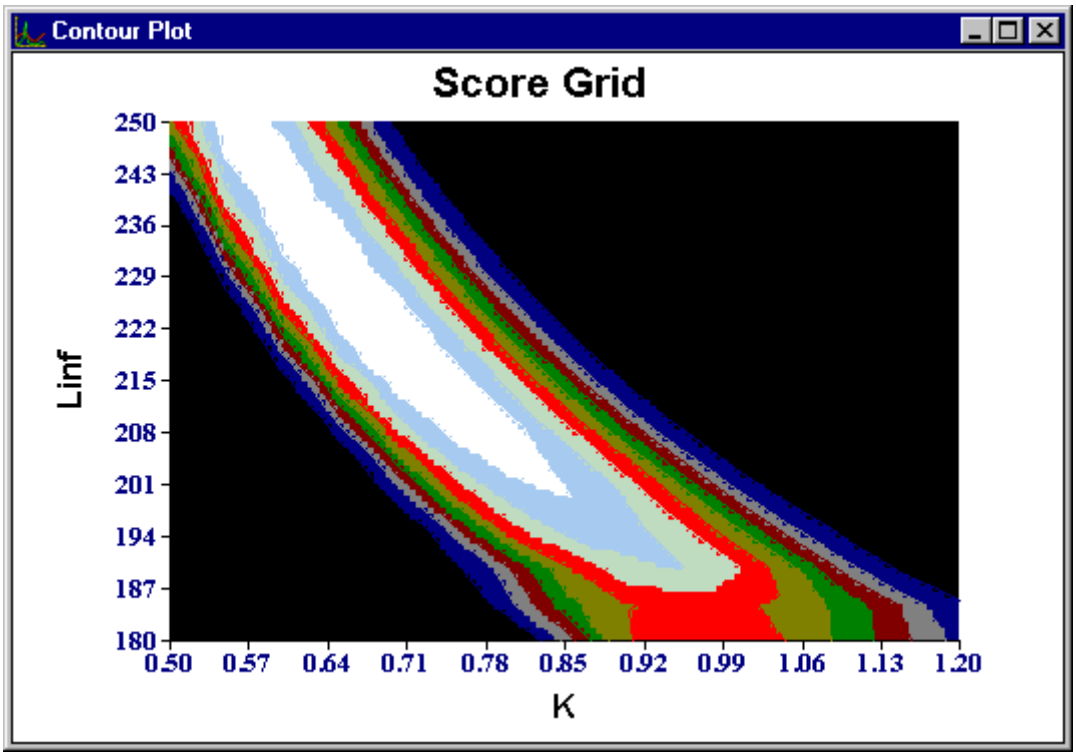


Figure 5.1.8 A Further Contour Plot Showing More Detail

You could continue the process of restricting the ranges of *K* and L_{∞} recalculating a table of score functions, and contouring and focusing until you finally identify the maximum, but that is not necessary. Instead, a procedure that attempts to find the maximum automatically is included in the LFDA package.

To invoke the maximisation procedure, return to the Parameter Estimates window (but do not close down the contour plot as this will return it to the default level of detail) and press the maximise button. This will bring up the Maximisation Options dialog box in which you are given a choice of using the 'current grid range' for the search or 'manually defining the search boundaries'. The 'current grid range' refers to the grid you defined last, and means that the maximisation routine would only look for the maximum score function within that range. For now you should select the 'current grid range option'. Below the grid range option is a check box which allows you to specify if you wish to view the maximisation process. You should usually make sure this box is checked as viewing the maximisation is useful. It will also help in explaining what the maximisation is doing, so you should make sure this option is checked for the purposes of the tutorial.

What happens next is that a progress bar appears and a lot of light blue lines are drawn all over the contour plot (Figure 5.1.9). What you are seeing is the maximisation routine searching for the point where the score function is a maximum. As a matter of fact, you are seeing five maximisation processes. Each maximisation run will generally find at least a local maximum, but the only way to be sure that you have found the global maximum is to start the maximisation process at different points within the grid range and then see if they all converge to the same maximum or, if not, which of the local maxima is the largest (refer to the Technical Appendix for more information). The starting point of each maximisation is marked with a yellow oval, except for the last one which starts in the middle of the search area. The path that the maximiser follows is drawn in light blue, and it stops once it has found a

maximum which is marked with a black rectangle. Once the last maximisation is complete, the biggest of the maxima is marked with a red rectangle.

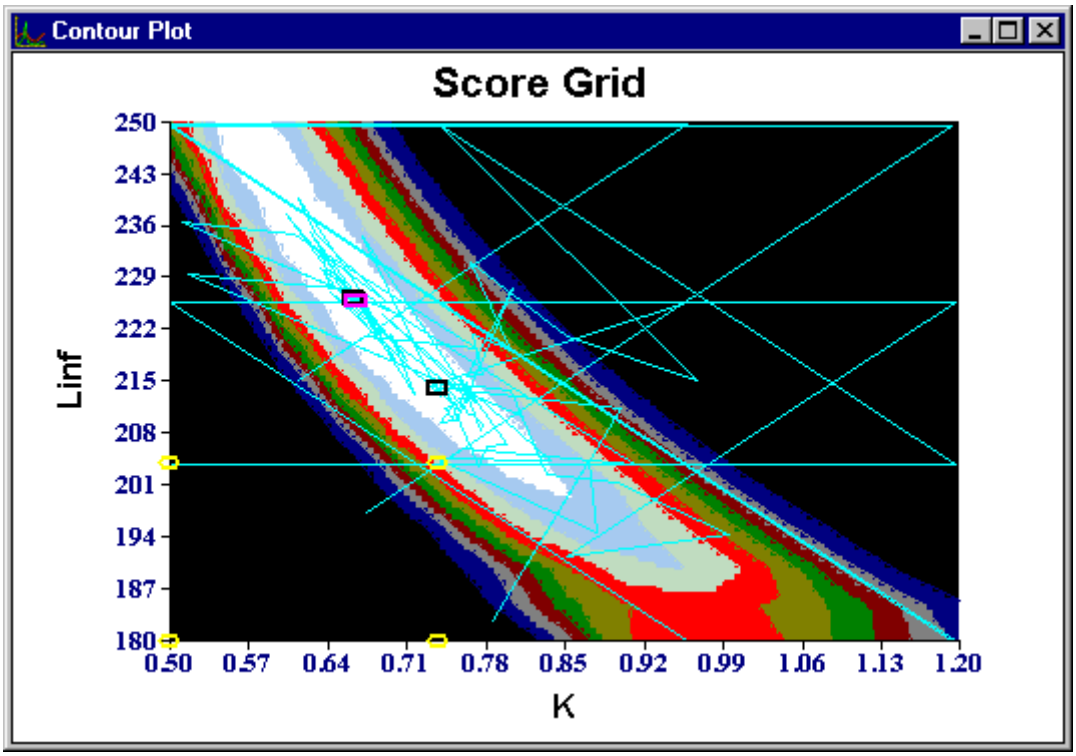


Figure 5.1.9 Contour plot redrawn showing steps leading to maximisation

In this case, you may have noticed that some of the lines converged to the point marked in red, but some stopped at a point slightly to the right and down. This latter must be a local maximum, but its score function value was still not as high as the other one and the parameter estimates are fairly similar. It is clear from the light blue lines that the maximisation routine searched over the whole white area, so it is reasonable to assume that the maximum it found was the true maximum for this method.

It is important to remember that this maximum represents the growth parameters that give the best fit of the von Bertalanffy growth curve to your data, using the SLCA score function as measure of 'goodness of fit'. How good is this fit really? To see the fit select the **Plot | Growth Curve** menu item from the Parameter Estimates window menu. LFDA will show the histogram plot of this growth curve superimposed upon the original length frequency data (Figure 5.1.10). As you can see, this fit is far from perfect. It will be interesting to see if any of the other methods can do better.

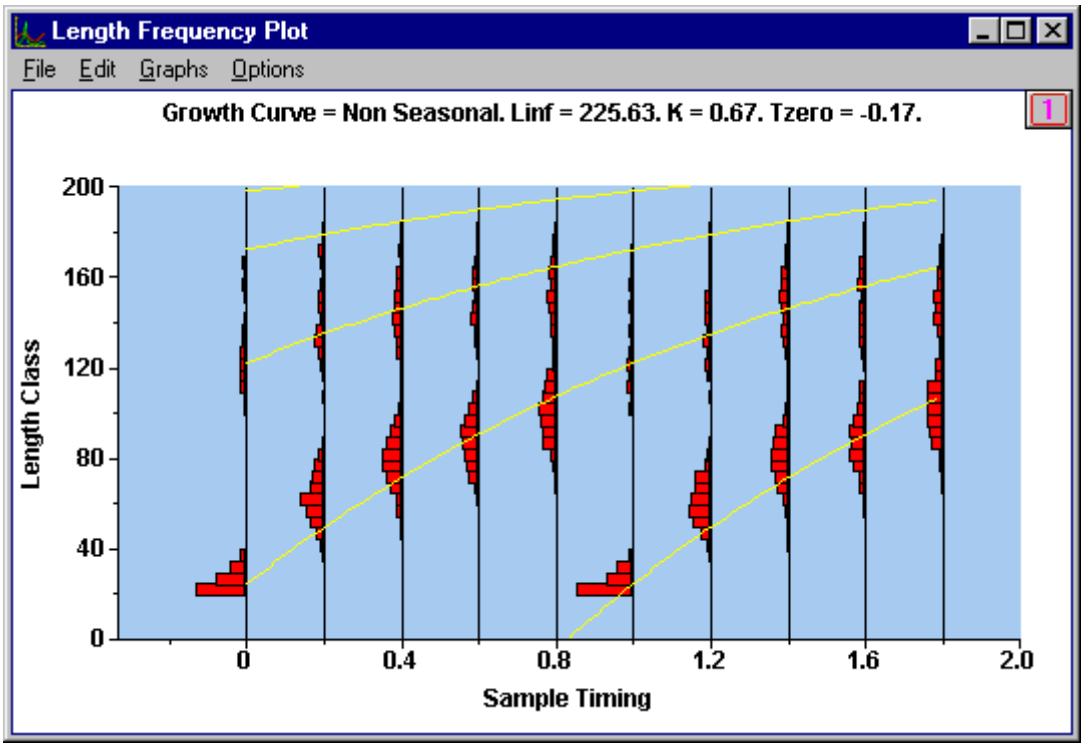
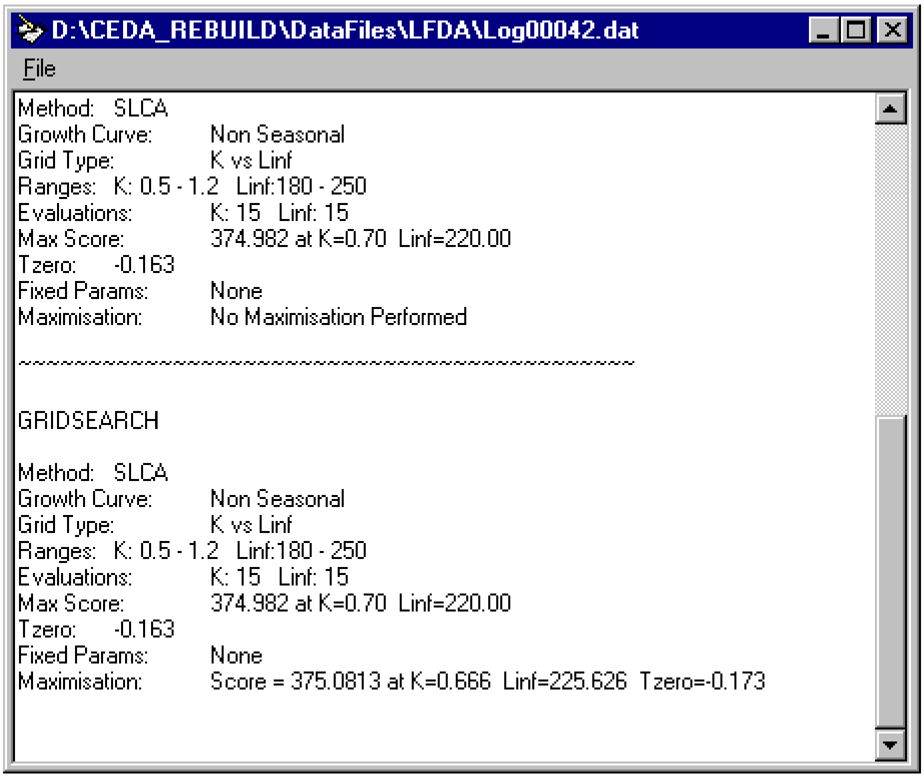


Figure 5.1.10 von Bertalanffy Growth curve fitted using SLCA score function

Returning our attention now to the Parameter Estimates window, the results panels look much as it did before, but this time there are maximisation results listed in the bottom panel : K=0.666, L_{∞} =225.63 and $t0$=-0.173.

A word of caution here: LFDA gives you quite a few decimal places to these parameters. This is to help you if the units of your data are not optimal, for instance if you are working in metres where centimetres would have been more appropriate. For data like these in the tutorial, this level of precision is not sensible, because you cannot hope that the results you get will be accurate to within 0.0005%! We will count ourselves lucky if we can get within 10% of the 'right answers' in this tutorial. Remember, a precise answer is not necessarily an accurate one.

LFDA keeps a copy of all the Parameter Estimate panels that you create during a session, and you can view this log at any stage to remind yourself what you have done. Let us look at it now. Close the plot of the growth curve if you wish to keep the desktop tidy and select the **Options | Log File** menu item from the main application window. You will see the Log window containing an ASCII text file (Figure 5.1.11).


D:\CEDA_REBUILD\DataFiles\LFDA\Log00042.dat

File

```
Method:   SLCA
Growth Curve:        Non Seasonal
Grid Type:           K vs Linf
Ranges:   K: 0.5 - 1.2   Linf:180 - 250
Evaluations:         K: 15   Linf: 15
Max Score:           374.982 at K=0.70  Linf=220.00
Tzero:      -0.163
Fixed Params:        None
Maximisation:        No Maximisation Performed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

GRIDSEARCH

Method:   SLCA
Growth Curve:        Non Seasonal
Grid Type:           K vs Linf
Ranges:   K: 0.5 - 1.2   Linf:180 - 250
Evaluations:         K: 15   Linf: 15
Max Score:           374.982 at K=0.70  Linf=220.00
Tzero:      -0.163
Fixed Params:        None
Maximisation:        Score = 375.0813 at K=0.666  Linf=225.626  Tzero=-0.173
```

Figure 5.1.12 View of the ASCII text Logfile

You can scroll up and down in this file, and besides a heading you should see the results of your two grid searches and the maximisation. When you leave LFDA you will be given the option of saving this Logfile. Note, you can save it at any time by using the **File | Save As** menu item in the Log window. Before moving on you must close the Log window.

The only other thing you might wish to do at this stage is to see what the parameters of the fitted growth curve indicate the age composition of your length frequencies might be. A means of doing this is included in the Functions menu. The method used is the so-called "age slice" method. As indicated in the Technical Appendix, this uses the growth curve to delineate the boundaries between lengths at age 0, 1, 2, etc., and then "slices" the length frequencies at those boundaries. Remember that because *t*0 is arbitrarily forced to lie between -1.0 and 0.0, the nominal ages might not be the true ages. Selecting the menu item **Functions | Age Slice** brings up a dialog box asking for growth parameters. Pressing the Calculate button will display a similar window to that we saw earlier for viewing the grid data. This window, with the caption "Age Slice Distributions" contains a table of estimates of numbers at age. From within this window you can save, print or copy the data to the windows clipboard using the appropriate icons on the tool bar. In order to plot the data you must return to the Age Slice Distributions dialog box and press the Plot button to produce Figure 5.1.12 below.

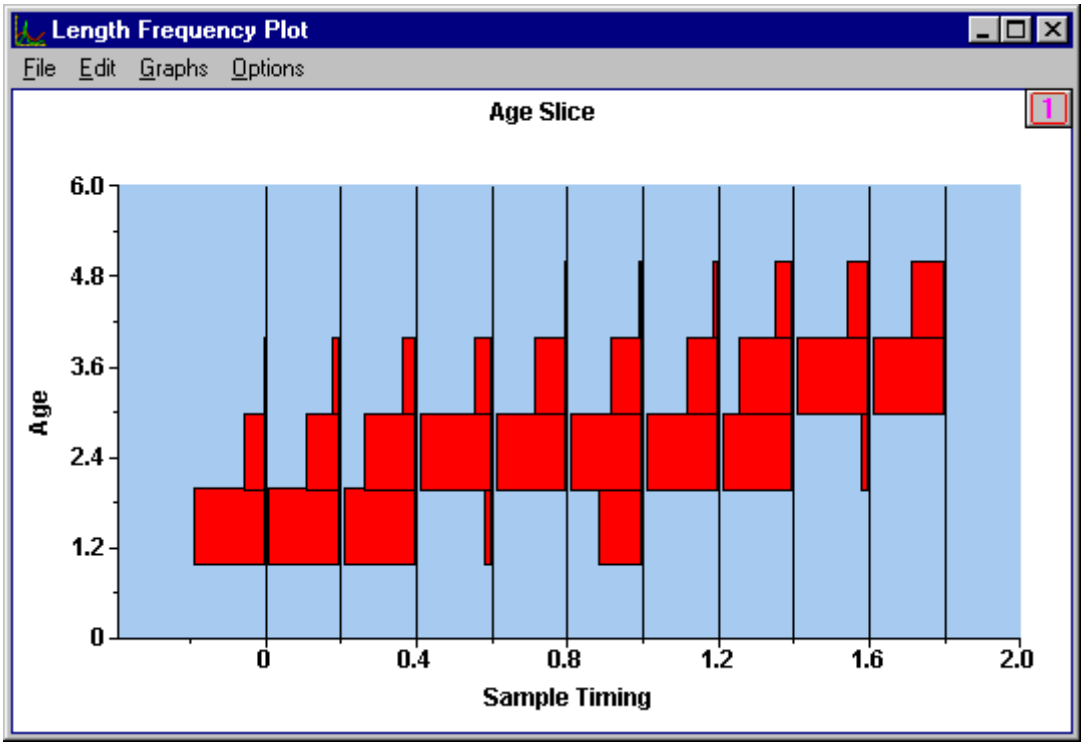


Figure 5.1.12 Age Slicing the length frequency data for each sample time

The next step is to see what parameters the other methods will estimate for the von Bertalanffy growth model. First close down all windows except the main application window and we will now move on to the Projection Matrix Method.

5.1.3.2 Projection Matrix Method (PROJMAT)

As explained elsewhere in the online help and in the Technical Appendix, the principles behind the projection matrix method for estimating growth curves are rather different from the other two methods, which rely more explicitly in fitting curves through modes. In the projection matrix method, successive pairs of length frequency distributions are compared. The basic idea is to successively project one observed length frequency distribution forward in time, based on an assumed set of growth curve parameters, to obtain a prediction of what that length frequency distribution should have looked like at the time the second observed length frequency was collected. The goodness of fit of the observed and predicted distributions are then compared. The best estimates of the growth parameters are those that lead to the best fit between the observed and predicted distributions.

Because the projection matrix method starts with an observed length frequency distribution and projects it forward over a given time interval, the method provides no information whatsoever on the parameter $t0$. This is the same thing that happens when you estimate growth parameters from tag-recapture experiments. However, in anticipation that you will normally also want an estimate of $t0$, we have provided estimates of that parameter for each pair of L_∞ and K values. These are calculated using the method incorporated in SLCA.

The mechanics of a PROJMAT analysis are pretty much the same as for SLCA. Let us start by seeing what PROJMAT makes of the grid we gave SLCA to start with: K from 0.1 to 1.5 in 15 steps and L_∞ from 150 to 250 in 11 steps. Select the menu item **Fit | Score Function Grid** to show the Grid Evaluation dialog box and this time change the method to PROJMAT. Leave the growth curve as it is, set to Non Seasonal. Now press the Define Grid Button and enter in the above grid specifications into the Grid Options dialog box. Select OK on the Grid Options dialog box and the Grid Evaluation dialog box to start computing the evaluations.

You will notice that the PROJMAT grid takes longer to calculate than the SLCA did. This is normal - in fact ELEFAN will probably take even longer. It is worth bearing this in mind if you are tempted to try large numbers of grid points.

The maximum value of the score function turns out to be -0.183 at K=1.30 and L_{∞} =160.00, with a corresponding SLCA-calculated $t0$ = -0.081. PROJMAT score functions are all negative, so the theoretical maximum score (for a perfect fit) would be zero.

Now draw the contour plot of this grid of values of K and L_{∞} (**Plot | Grid Data**). A bit more detail would be useful here. Pressing the right hand mouse button once over the grid will display the contour plot illustrated in Figure 5.1.13.

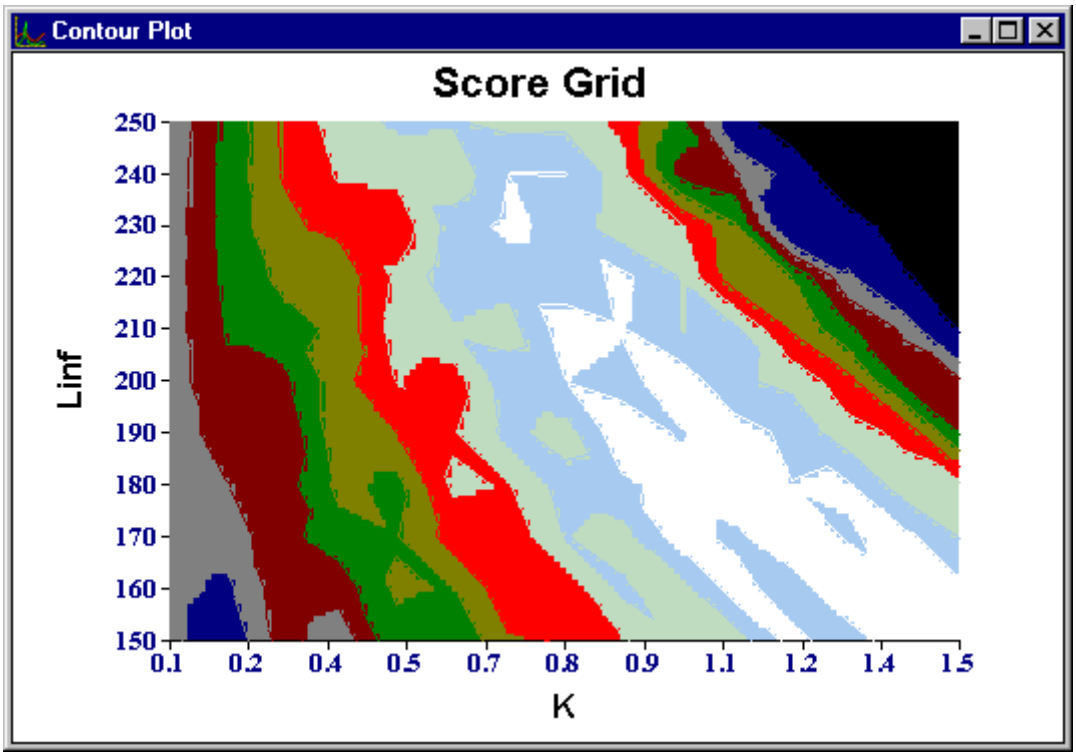


Figure 5.1.13 Contour Plot for PROJMAT score function

An immediately noticeable feature of this plot is that we have a number of disjointed regions where the maximum could be. We may need to adapt our search strategy to examine each of these regions in turn. As a first step, though, it seems clear that our grid included K's that were too low - a bottom limit of 0.7 would be more reasonable. It does seem as if higher K's and L_{∞}'s might be possible, although, as before, an L_{∞} below 150 seems unlikely. We will neglect these combinations unless we find that the score function maxima come close to this edge.

Let us quickly recalculate the grid, but with K from 0.7 to 1.5 in 17 steps. You will see that the maximum score function this time is at L_{∞} =150, which is on the edge of our grid. This means that our grid is not large enough - the maximum we are looking for may well be off the screen. Accordingly, we had better enlarge the grid to cover K from 0.7 to 1.7 in 21 steps and L_{∞} from 130 to 250 in 13 steps. The maximum is still at L_{∞} =150 and K=1.45, but at least this time we know that it is not at any lower values of L_{∞}. If you redraw the contour plot and magnify it once, you will see that the white area is far more fragmented than it was for SLCA. Our grid seems to be well defined, though, so we can now see what the automatic maximisation finds.

Do not close down the contour plot window and perform a maximisation with the view maximisation option checked to see the plot below (Figure 5.1.14).

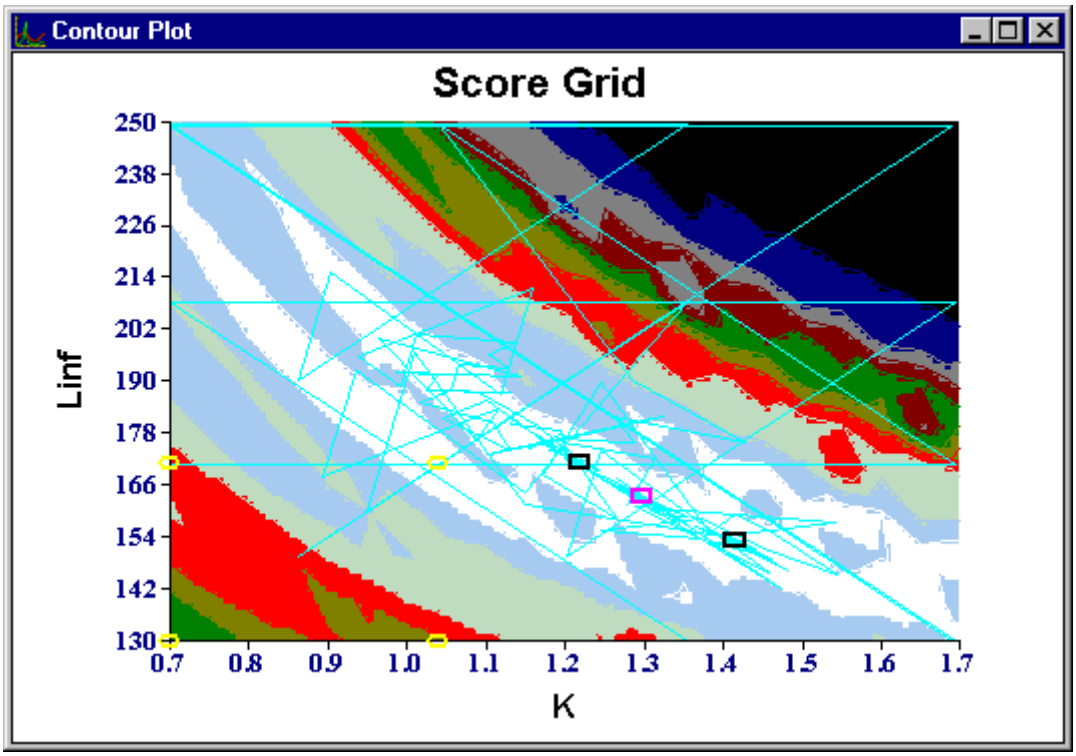


Figure 5.1.14 Contour Plot With Maximisation for PROJMAT

The white area in the contour plot can be seen as two banana shaped strips lying one above the other. The maximiser has found three local maxima in the uppermost strip, but only seems to have tested the lower strip at the top left and bottom right extremes (corner points of the red lines are where the score function has been tested). There may be another maximum here that has escaped notice, so we will have to focus on this area separately. First, we must make a note of the maxima that have been found. The global maximum (the pink rectangle) was the score function value -0.176 at K=1.298, L_{∞} =162.997 and $t0$=-0.066. The other maxima were found at around K=1.22 and L_{∞} =170 and K = 1.42 and L_{∞} =154. It is useful to note these secondary maxima because they may coincide with maxima found using the other methods. As it is, these parameter estimates are rather far from the values estimated by SLCA.

The next step is to check for potential maxima in the lower of the two white strips. Close down the contour plot and return to the Parameter Estimates window. We need a new contour plot at the correct level of detail for the maximisation process. Therefore, produce a plot of the contour data and right click on it once to produce the same level of detail we had before. We will now proceed with maximising over the lower of the two white strips. Press the Maximisation button. This time, select the 'Manually define search boundaries' option. Text boxes for entering the new ranges will become enabled. Type in the new ranges K from 0.9 to 1.1 and L_{∞} from 150 to 190. Make sure the 'View Maximisation' option is checked and press OK to maxise. You will now see a set of five maximisations on the contour plot, but this time they are confined to the area we specified. The score function is maximised at K=0.974, L_{∞} =178.79 and $t0$=-0.14 with a score function value of -0.192. This score function value is quite a bit lower than the ones we found earlier, so the other, earlier estimates are still the best ones obtained with the PROJMAT method. Before we pronounce this finally, though, it is worth comparing the model fits obtained with these two sets of parameters. Remove the contour plot and from the main application window select **Data | Plot Fit** which will plot the growth curve using the parameters from the last maximisation we did. The graph window has

a series of menu items for printing the graph page, copying it to the windows clipboard or saving it in Windows metafile format. You should print this plot now by selecting **File | Print Current Page** from the graph window.

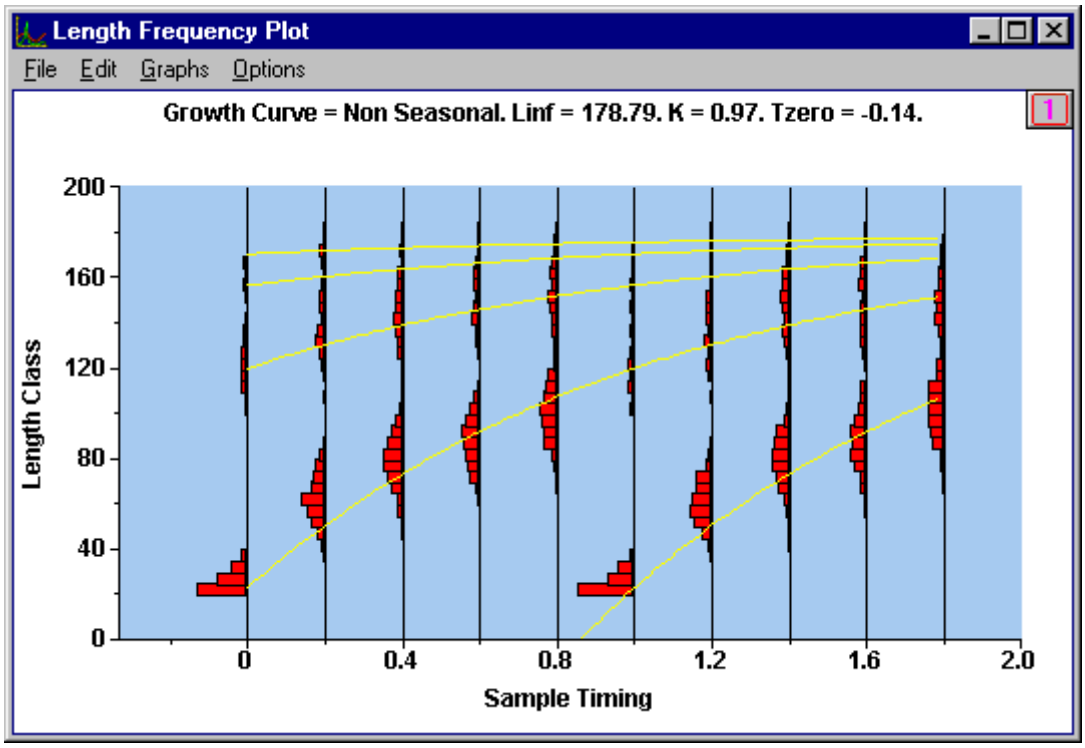


Fig 5.1.15 a Fitted growth curve for first set of PROJMAT estimates

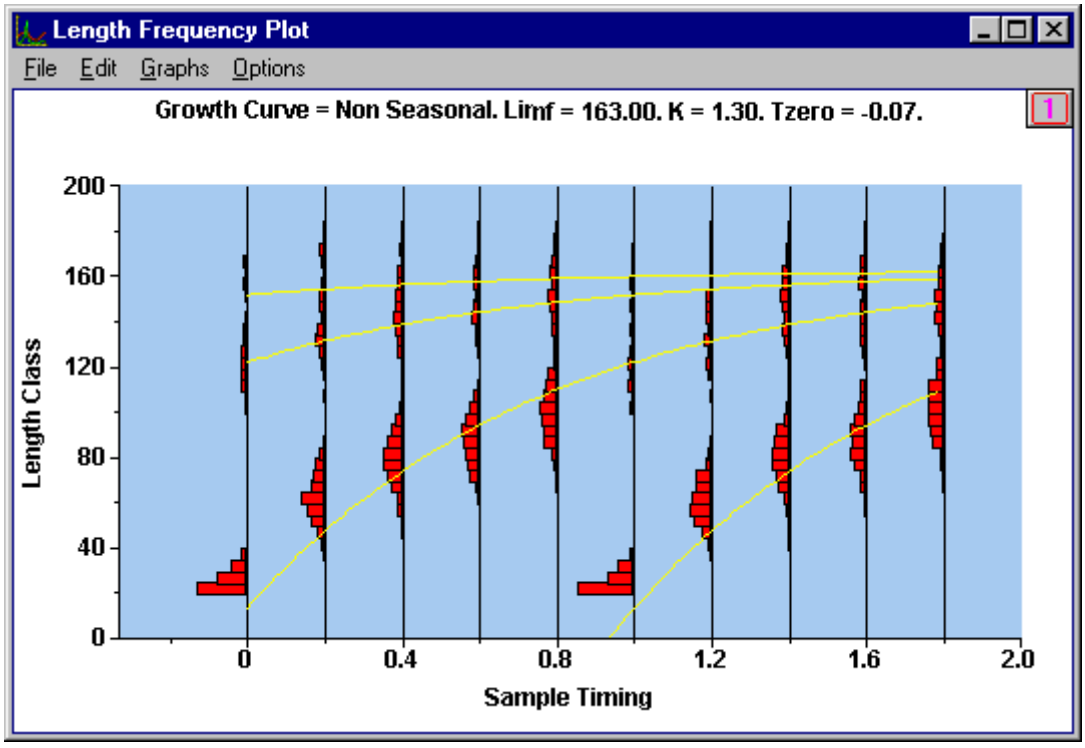


Fig 5.1.15 b Fitted growth curve for second set of PROJMAT estimates

Next, repeat the above plotting process, filling in the best parameters we found earlier, that is K=1.298, L_{∞} =162.997 and $t0$ = -0.066. Print this plot as well to compare the two fits (Figure 5.1.15 a and b). This is important because sometimes PROJMAT can make a mistake and miss out a complete cohort. If something this drastic is happening you will see it clearly on the **Data | Plot Fit**. In this case there is no obvious problem of that sort. The fits seem quite similar for low ages and although they are different nearer the asymptote it is clear that they are aiming at different peaks within the higher length class data. There is no way of choosing one above the other as yet.

As a final check, let us look for a maximum specifically in the region where SLCA produced its best estimates. Repeat the maximisation process but use the search range K from 0.7 to 1.1 and L_{∞} from 190 to 250. You will see that all the maxima are on the boundary of the region at L_{∞} =190. This means that in searching for the maximum the program kept running into this edge and then being forced to stop. This is a clear indication that the maximum lies outside this region. Remember that the score functions of SLCA and PROJMAT are quite different, so the fact that they have produced such different parameter estimates is unfortunate but not a contradiction.

5.1.3.3 ELEFAN Method

The third method for estimating von Bertalanffy growth parameters included in the LFDA package is based on the ELEFAN 1 method devised by Dr Daniel Pauly and reviewed in Pauly (1987). It works by first restructuring the length frequency data in a way that emphasises the peaks and troughs in the data, and calculates a score function as a function of the proportion of available peaks and troughs that can be explained by a von Bertalanffy growth curve with specified parameters. See the Technical Appendix for more detail of the method and the modifications included in the LFDA package.

To start the process, first calculate the values of the ELEFAN score function (remember to change the model to ELEFAN) for the now familiar initial range K=0.1 - 1.5 step 15 and L_{∞}=150 - 250 step 11. ELEFAN starts calculating very slowly, but speeds up as it proceeds. You should find that the maximum amongst the grid of values calculated is 0.453, and it is found where K=0.5, L_{∞} =200 and $t0$ = -0.65. This time, though, the plot of the fitted von Bertalanffy curve looks quite different because it is superimposed on the “restructured length frequency data” rather than the original data. You should see a plot similar to Figure 5.1.16

The process of restructuring is specified fully in the Reference and Operating guide, while ELEFAN is also described in the Technical Appendix. Basically, the restructuring emphasises the modes in the data, and the solid histograms in the restructured plot represent the clearest peaks in the original data. You will see groups of solid red bars separated by white ones, where each group of solid red bars represents a cohort (Figure 5.1.16).

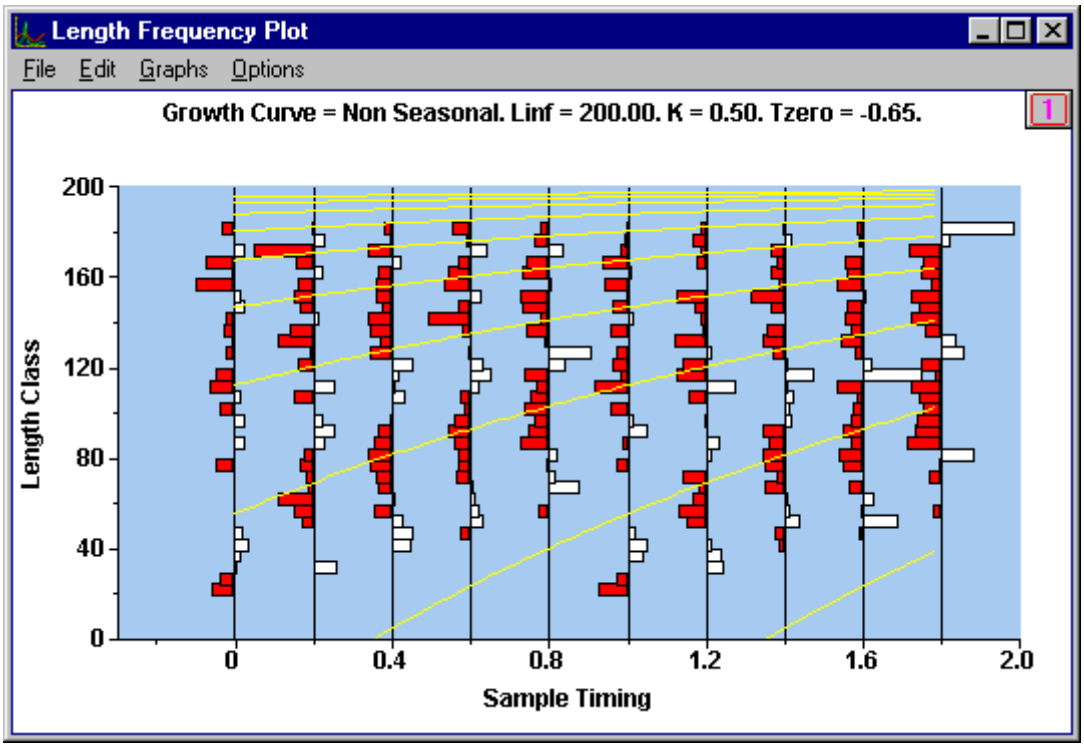


Figure 5.1.16 von Bertalanffy Growth Curve Fitted By ELEFAN

ELEFAN will try to fit a growth curve through the base of the highest one of the solid bars in each group. As you can see, it is not doing too badly, although there is plenty of scope for improvement. You can redraw this plot at any later time by selecting **Fit | Plot Restructured Data** on the main application window. You can also see an ELEFAN fit plotted in the conventional way by selecting **Data | Plot Fit.**

Now generate a contour plot. After one click on the right hand mouse button to focus, the plot will look like Figure 5.1.17.

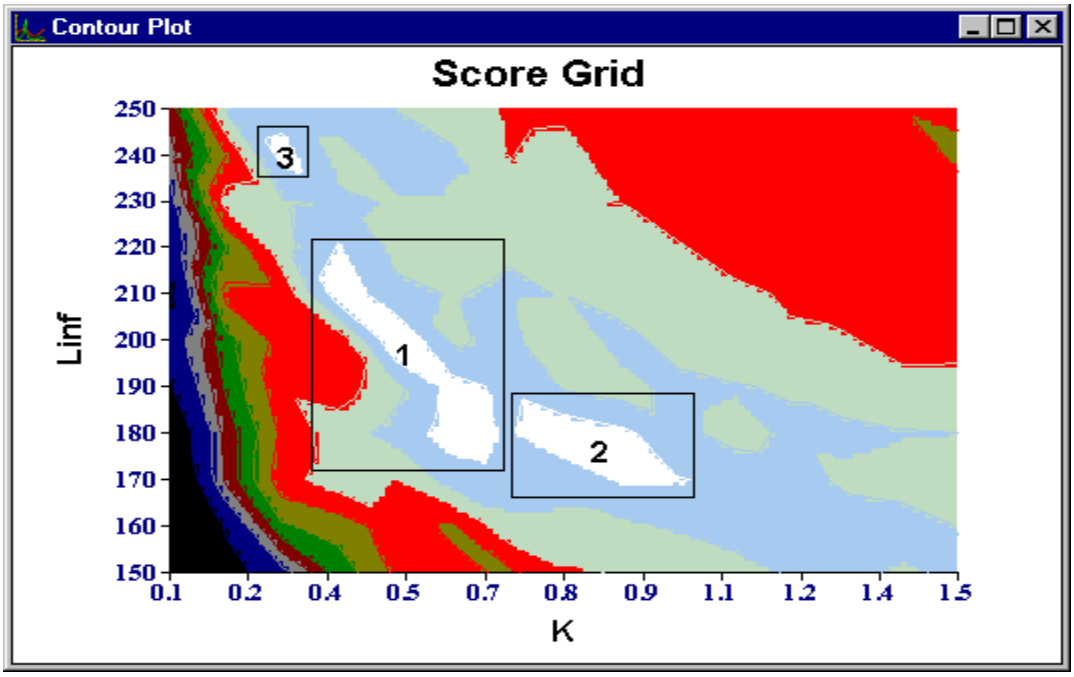


Figure 5.1.17 Contour Plot for ELEFAN score function showing the three regions to be examined separately

You will notice that the total white area seems smaller than was the case with the other two methods. The stronger the modes in your data, the more this will be the case when using ELEFAN. We will be able to narrow down our ranges considerably, but given how fragmented the white areas are, we may need to examine a few of the regions separately. The major white areas fall within three main regions which are marked in Figure 5.1.17. These regions are small enough that we can try to run the maximiser bounded to each of these regions in turn. The bounds of the regions are given below:

Region	*K* Bounds	L_{∞} Bounds
1	0.37 - 0.72	173 - 220
2	0.73 - 1.02	170 - 188
3	0.27 - 0.35	235 - 245

First, let us search for the maximum over the whole grid. If you do this you will see that the two lower areas of white are searched quite thoroughly and the maximum is where the search function equals 0.466, at K=0.841, L_{∞}=180.51 and $t0$=-0.16. The results of searching in all three regions separately are shown in the table below:

Region	Score	K	L_{∞}	T_0
1	0.469	0.502	198.264	-0.670
2	0.466	0.841	180.51	-0.160
3	0.405	0.297	241.000	-0.120

The score function for the third region is much lower than the other two, with parameters different to anything else obtained so far, so we need not consider this point further. Let us have another look at the maxima from the first and second regions. The score functions are very close to each other, but the estimated parameters are quite different. We need to look at the fit plots to understand what is happening here. Use **Data | Plot Fit** to plot the fits corresponding to the two sets of parameters, and print both plots so that you can compare them. They should look like Figure 5.1.18 a and b below, and as you can see, the fitted lines are quite different.

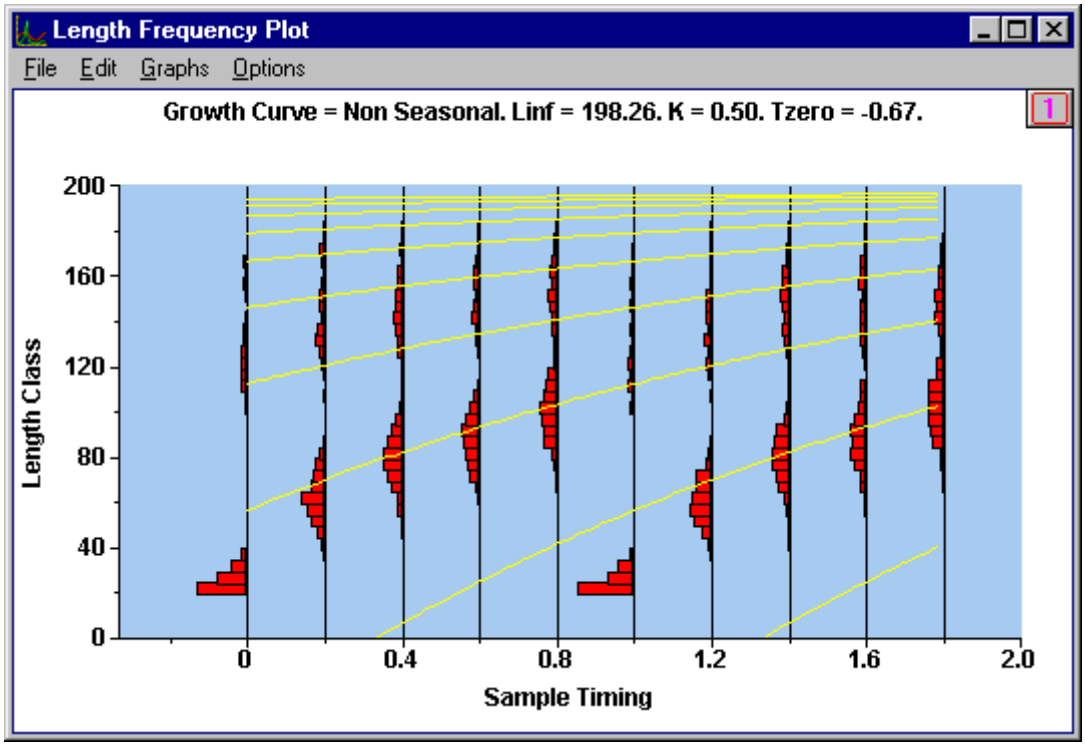


Figure 5.1.18 a Growth Curve using parameters estimated from Region 1

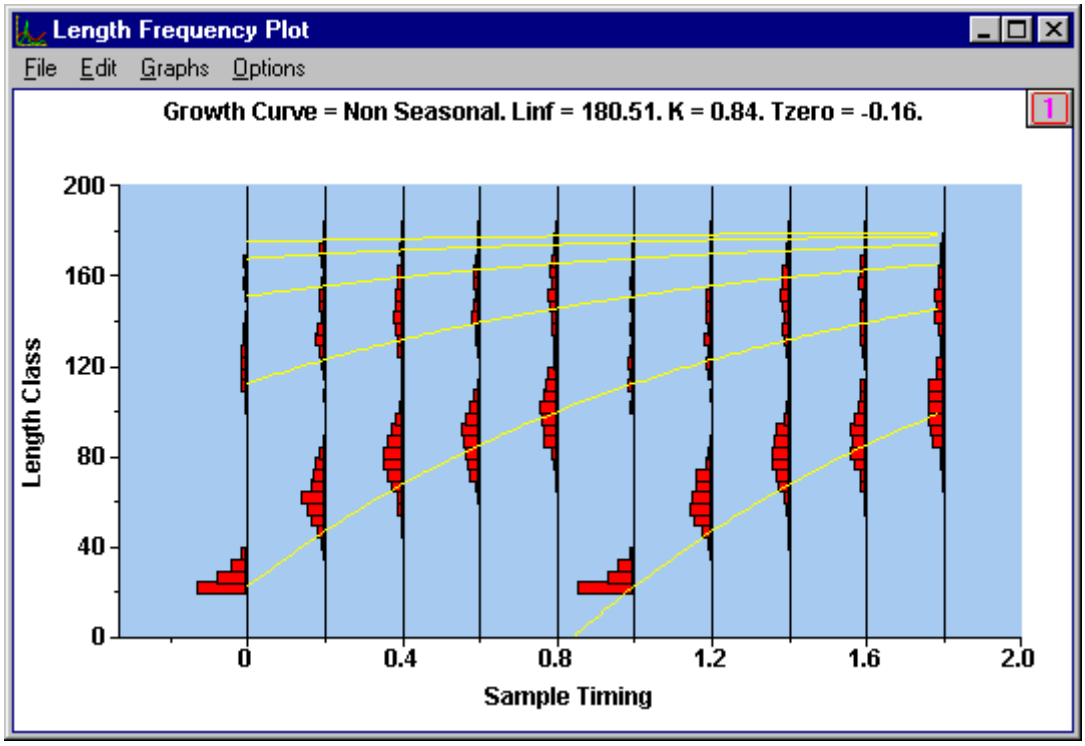


Figure 5.1.18b Growth Curve using parameters estimated from Region 2

What seems to be happening is that the *K*=0.502 (Figure 5.1.18, Plot a) curve manages to fit the end-of-year data fairly well, but is bad early in the year and gives up totally on the first mode. The *K*=0.841 (Figure 5.1.18, Plot b) option is good early and late in the year but misses completely in the middle. If you were to look again at PROJMAT's best estimate at *K*=1.3, you would see a similar pattern. It is as if the data can fit the von Bertalanffy curve in places, but not everywhere. More importantly, it misses the curve consistently across cohorts at particular times of the year. This suggests that the stock may be displaying a seasonal growth pattern. We will examine this possibility in the next section, Estimation of Seasonal Growth Parameters..

5.1.4 Estimation of seasonal growth parameters

In the previous section we completed a thorough attempt to fit a von Bertalanffy growth model to our data. Normally, one would hope to find reasonable correlation between the parameter estimates provided by the different methods. In this case, however, we suspect that the stock may be growing faster at different times of the year. This sort of growth is described by the Hoenig and Pauly functions. Pauly's model is specifically geared for species that stop growing completely at certain times of the year, like Atlantic salmon. This is a more specialised model, and we have no reason to suspect that our stock may be behaving in this way. So, our first attempt will be to fit the more general Hoenig function. It is important to remember that one should have good biological reasons for trying a seasonal growth curve. Just getting a better score function with one is not good enough. Most datasets will give you a better score function with a Hoenig model than a von Bertalanffy, simply because the extra parameters can explain away some of the random variation in the data.

It will be useful to have a summary of the von Bertalanffy parameter estimates obtained so far, to give us some idea of where to start looking (see below). Unfortunately, these do seem to cover a wide range, but we think we know why this is happening and we hope to do better once we fit a more realistic model.

Estimation Method	L_∞	*K*	*t zero*	Score Function
SLCA	225.63	0.666	-0.173	375.081
PROJMAT	162.977	1.298	-0.08	-0.176
ELEFAN	198.264	0.502	-0.670	0.469
	180.51	0.841	-0.160	0.466

5.1.4.1 PROJMAT/Hoenig

We will start with the PROJMAT method, making use of the maxima that we have found so far. Select Fit **| Score Function Grid** to define a new grid, but this time select the PROJMAT method, the Hoenig model, and choose the grid type *C vs Ts*. We will need to estimate *C* and *Ts* as well as *K* and $L_¥$, and we do this a pair at a time. In the Grid Options Dialog Box we must specify the grid for *C* and *Ts*. *C* always takes a value between 0 and 1, and *Ts* must lie between -0.5 and 0.5. These are therefore the default grid boundaries and you can usually just accept these. You must also specify values for *K* and $L_¥$; - these are fixed values and will remain constant while we are estimating *C* and *Ts*. Enter the best estimates for the von Bertalanffy curve that we arrived at using PROJMAT, i.e. *K*=1.298 and $L_¥$ =162.977. Press OK to accept these values and OK once more to start calculating the *C vs Ts* grid.

As before, the best grid point will be identified, and this time it is at *C*=0.71 and *Ts*=0. The score function is -0.110, which is already much better than the previous PROJMAT best at -0.176. (Remember that the maximum value of a negative score function is the one with the smallest absolute value). The plot of the fit gives you your first glimpse of a seasonal model at work. The contour plot for this grid looks quite different (Figure 5.1.19). The next step is to maximise the function as before, by pressing the maximise button. It can help, for these grids, to use smaller boundaries for the maximisation, because the program can sometimes be distracted by spurious local maxima on the borders. It is easy to see from the contour plot where the boundaries can safely be drawn. Try 0.1<*C*<0.9 and -0.2<*Ts*<0.2. Once you have found the maximum, you will see that there has been a further improvement in the score function, now at -0.097. This corresponds to the parameters *C*=0.505 and *Ts*=-0.018. If you confine the maximisation to the right half of the above area, you will see that the maximum there is lower than this one.

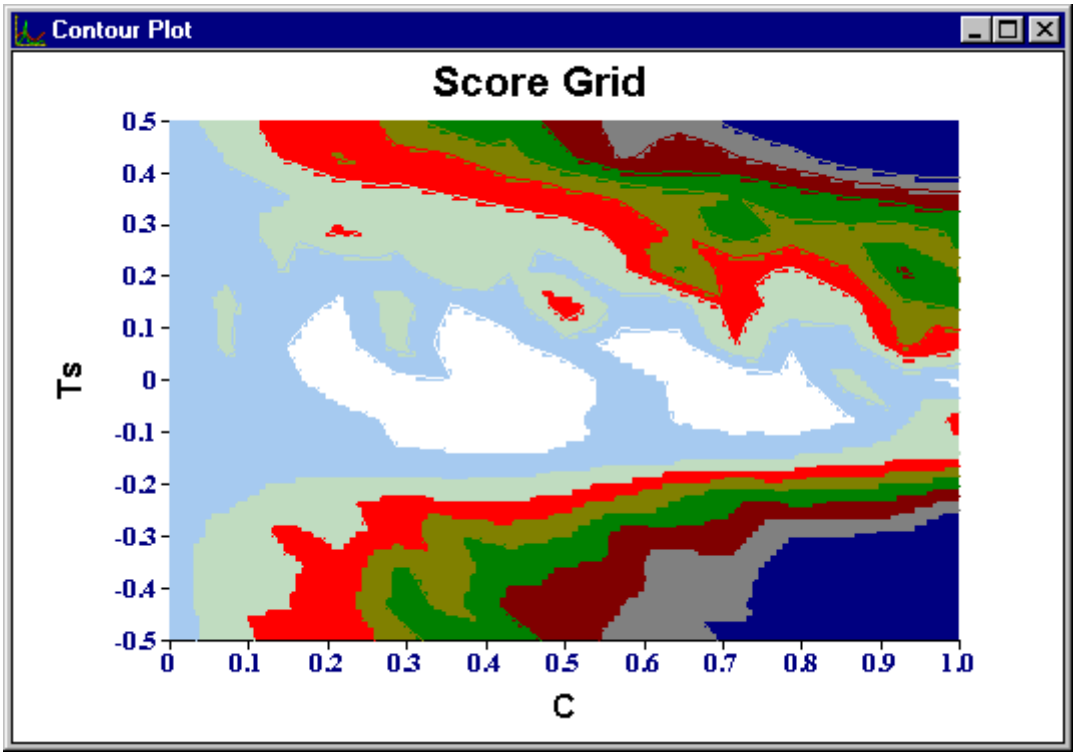


Figure 5.1.19 PROJMAT / Hoenig Contour Plot

Let us accept these parameters for a start and see if we can find better estimates of *K* and $L_{¥}$, now that we have new values for *C* and *Ts*. (A von Bertalanffy curve is like a Hoenig curve with *C*=0 and *Ts*=0). Define a new *K vs* $L_{¥}$ grid using boundaries spread around the previous estimates, for example K from 1.1 to 1.5 in 13 steps and $L_{¥}$ from 140 to 180 in 13 steps. You will be prompted for the fixed parameters *C* and *Ts*, while the last estimated values will have been filled in as defaults so you can accept these. Now, surprisingly, we find that the maximum is towards the corner of the grid, at *K*=1.1 and $L_{¥}$ =166.67. Changing *C* and *Ts* has affected *K* and $L_{¥}$ more that we expected. At least we know in which direction the score function maximum lies. A logical next step would be *K* between 0.9 and 1.3, and $L_{¥}$ between 150 and 190. This is only a slight improvement, giving *K*=0.97 and $L_{¥}$ =180, which is still on one edge of the grid. We know that we are on the right track, however, because the score function is still increasing, and currently stands at -0.083.

Interestingly enough, the point we are at now corresponds quite closely to the smaller local maximum that PROJMAT identified for the von Bertalanffy model. If this had not happened, we would have had to investigate that other point separately, but it seems that the analysis we are doing will serve for both points.

We need to shift the grid again, so this time try *K* from 0.8 to 1.2 in 13 steps and $L_{¥}$ from 160 to 200 in 13 steps. This time the maximum is comfortably in the middle of the grid, at *K*=0.9 and $L_{¥}$ =193.33 with a score function of -0.081. Once again the contour plot has two strips of white separated by an area of lower score function values, so we need to make sure any maximisation examines both areas (Figure 5.1.20). Do an automatic maximisation using the grid range as the search boundaries. You will see that the lower strip is investigated quite thoroughly, but the light blue lines hardly touch the upper strip at all. The maximum in the lower strip has a score function value of -0.081, for *K*=0.902 and $L_{¥}$ =193.48 and *to*=-0.13. If we repeat the maximisation, restricting the search to *K*>1, we find maxima in the upper strip but the best has a function value of only -0.093 which is further from zero than our first result. You may notice that the maximisation is quite sensitive to search boundaries and values for fixed parameters, so on your own data it is worth trying a number of such variations.

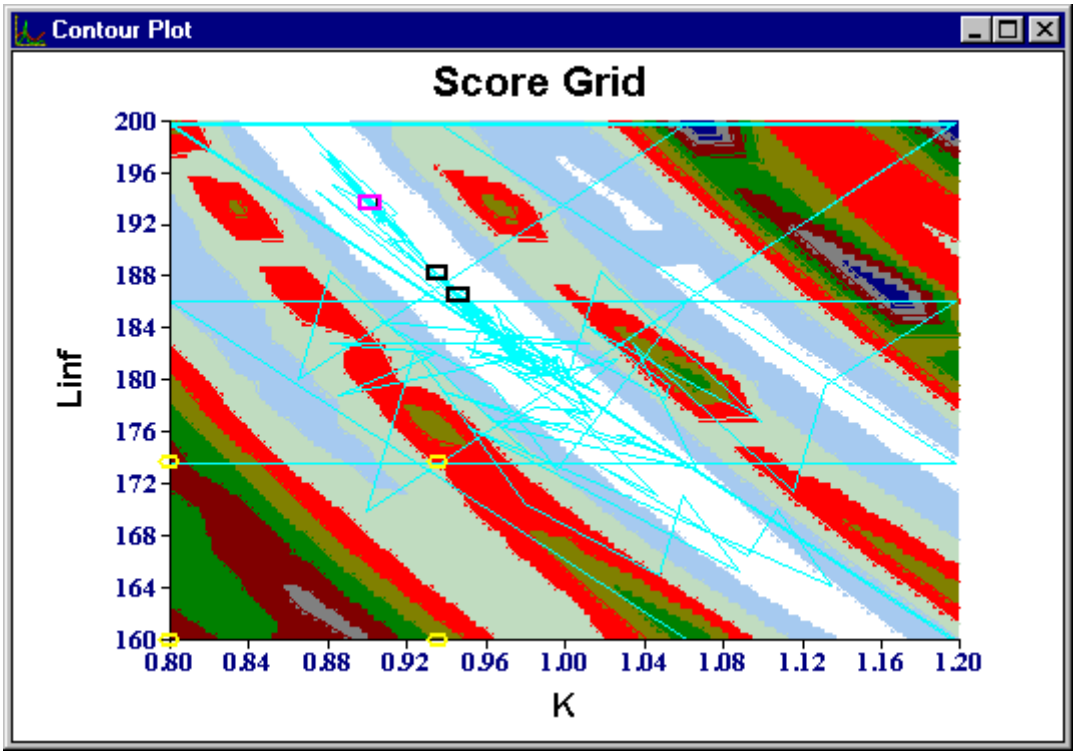


Figure 5.1.20 PROJMAT / Hoenig Contour Plot with Maximisation

Now that we are happy with our new estimates for *K* and L_{∞}, we can try to refine *C* and *Ts*. Define a new *C vs Ts* grid, accepting the default ranges once more. When you are prompted for constant values for *K* and L_{∞}, you will see that the last estimates you obtained are filled in for you. In this case we are not interested in the last maximisation that we did, so fill in our best estimates *K*=0.902 and L_{∞}=193.48. Once the grid has been calculated and you have looked at the contour plot, run a maximisation again with a restricted range. This time, you should find *C*=0.6.7 and *Ts*=0.029 to provide the best score function of -0.078.

Return now to a *K vs* L_{∞} grid, defined for *K* from 0.7 to 1 and L_{∞} from 180 to 210, in 11 and 13 steps respectively. The best grid score function is -0.074 which maximises to a score function value of -0.074 at *K*=0.76, L_{∞} =200.00 and *t*0=-0.14.

If you continue repeating the process of maximising for *C* and *Ts*, then using those values to maximise for *K* and L_{∞}, etc, you might be able to improve the score function even more. However, it does seem as if we are settling at values around *K*=0.75, L_{∞} =200 and *t*0=-0.15. If you plot the histogram fit with these values you will see that it really looks quite good.

You may have noticed the menu item Fit **| Seasonal Growth Curve**. This function performs the cycle that we have been doing here, automatically. There is a limit to how much you can expect from such a routine, though, so you should only use this when you have a good idea of what *K* or L_{∞} is, and can specify it very closely. It is not uncommon for L_{∞} to be known by other means, and then you would be justified in using **Fit | Seasonal Growth Curve**. In general, the manual method is far more effective because you can apply your own intelligence to the problem; you can backtrack and try other things where the automatic routine would proceed blindly. You will see this sort of thing happening in the next section, where we see what ELEFAN can do with our data.

5.1.4.2 ELEFAN / Hoenig

In our earlier use of ELEFAN to fit a von Bertalanffy model, we ended up with two fits with very similar score functions but quite different growth parameters. These were close enough to warrant separate investigation.

We will start with the best maximum we found, which was for Region 1 (Figure 5.1.18) at K=0.502 and L_∞=198.264. The first cycle should estimate C=0.190 and Ts=-0.147 with a score function of 0.488 which is already much better than the 0.469 that we started with. Returning to estimate K and L_∞ using the grid $0.4<K<0.7$ and $180< L_\infty <220$ shows us that the grid was far wider than necessary. We can maximise over the smaller area $0.4<K<0.6$ and $190< L_\infty <210$ to obtain a score function value of 0.492, our highest yet. You will find that repeating the cycle gives C and Ts values very close to the previous ones. The K and L_∞ estimates also do not change much, so all seems to stabilise at K=0.49, L_∞=200.00, to=-0.7, C=0.19 and Ts=-0.15, with score function value 0.492.

Now let us examine the second maximum; although it was smaller than the first it is different enough to be worth checking. Looking at this pays off quickly, too, because the C vs Ts grid produces a score function value of 0.501, which is already better than our final value from the first point. Maximising over this grid produces another surprise: the global maximum is at C=0.358 and Ts=0.231. If we calculate another K vs L_∞ grid over a reasonable range like K=0.6 - 1 and L_∞=160 - 200 we can then maximise over an appropriate smaller area, in this case. The maximum is at K=0.810, L_∞=181.568 and to=-0.24 with a score function value of 0.516. If we calculate a new C vs Ts grid, we will find that we can maximise over the smaller range C=0.2 - 0.6 and Ts=0.1 - 0.4. The score function increases to 0.518 for the estimates K=0.811, L_∞=181.502, to=-0.24, C=0.412 and Ts=0.253.

As before, you may be able to improve on this by repeating the cycle. If the parameters do not change much anymore, it is not worth pursuing further. Remember always that you cannot expect a high degree of accuracy from estimation methods, so there is no sense in trying to estimate an answer that is precise to two decimal places.

5.1.4.3 Pauly's model

If we had estimated C to be near to or greater than 1, we might have tried to use Pauly's model. This is basically like a Hoenig model with C=1 and a period of the year (*NGT*) in which no growth occurs. You should not try to fit a Pauly model unless you have good biological reason to suspect that your stock experiences such suspended growth in certain seasons.

Fitting such a curve involves the same process as for a Hoenig model, only in this case you cycle between *K vs* L_∞ grids and *NGT vs Ts* grids. If are already fairly certain of L_∞ then you could try using the *K vs NGT* grid instead. *Ts* can be used to adjust the time of year at which the no-growth period occurs, while *NGT* is the length of that time. This subject will not be covered further in this tutorial, but it is discussed fully in the Technical Appendix.

5.1.4.4 Comparison of estimates of seasonal growth parameters

We have now used two methods for estimating the Hoenig growth parameters L_∞, K, $t0$, C and Ts. How well do the estimates compare, both with each other and with the "true" values used to simulate our data? This is shown in the table below, and the fits are compared in Figure 5.1.21 a and b.

Method	L_{∞}	*K*	*t0*	*C*	*Ts*
PROJMAT	200.00	0.76	-0.14	0.64	0.03
ELEFAN	181.50	0.811	-0.24	0.412	0.253
"True" values	175	1.0	-0.1	0.6	0.0

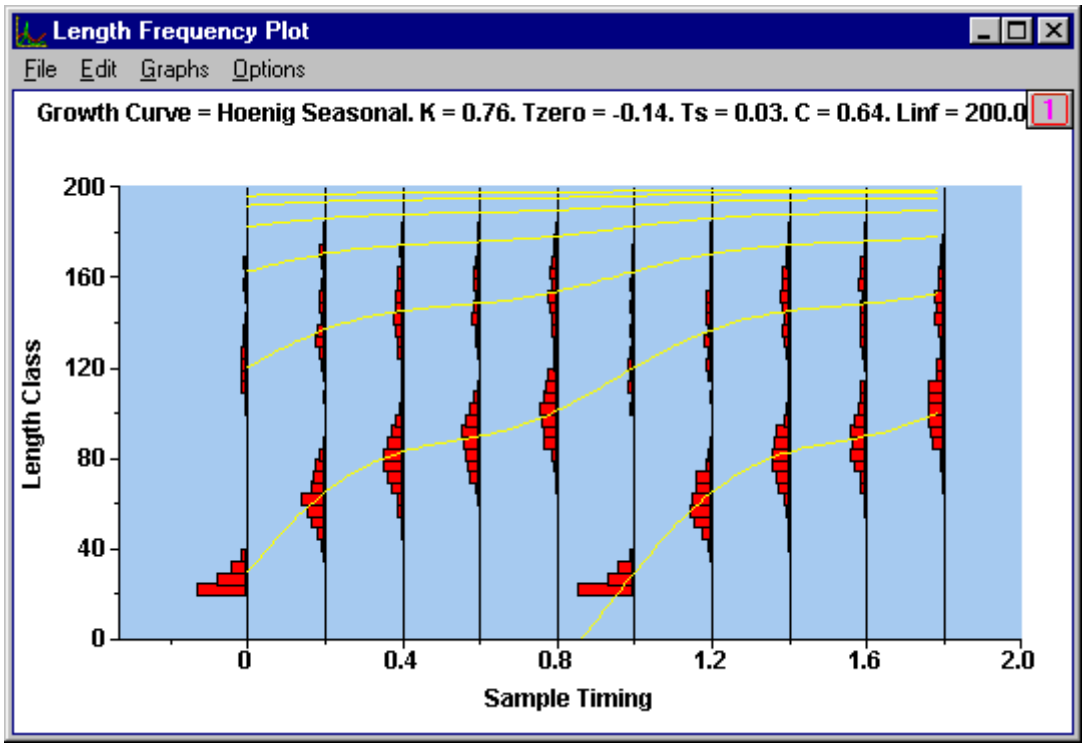


Fig 5.1.21 a. Estimates of seasonal growth parameters using PROJMAT

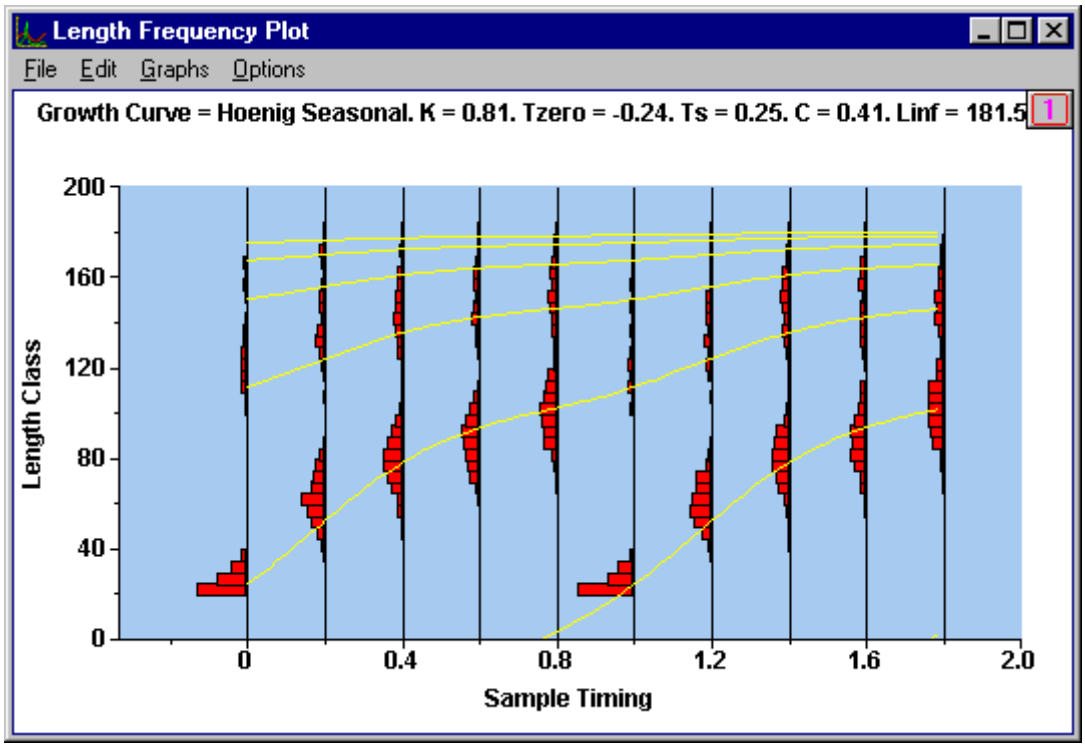


Fig 5.1.21 b Estimates of seasonal growth parameters using ELEFAN

ELEFAN has come the closest to K and L_{∞} (Figure 5.1.21 b), having estimates within 20% of the true values. PROJMAT has done much better with *C*, *t0* and *Ts* (Figure 5.1.21 a). Note that you cannot get a better score function value by using the "true" parameters. The discrepancy is not because we did not look in the right place, but rather because of the inherent problems involved in this sort of estimation. No method and score function is perfect, which is why we try a number of different ones. In the end, our two methods produced answers that were close enough to each other to give us reasonable confidence in the order of magnitude of our estimates. It would be unwise to expect more from your analysis of your own data.

5.1.5 Estimating total mortality rates Z

Having obtained estimates of growth parameters, we now turn to estimating the total mortality rate, Z. Three methods are available in the LFDA package. Unfortunately, these methods are all based on non-seasonal von Bertalanffy growth curves, so cannot be used to estimate mortality for a stock displaying strongly seasonal growth. This is not normally a problem since most stocks' growth can be reasonably described by the non-seasonal von Bertalanffy model. The case of strongly seasonal growth is a difficult one, for it is unlikely that such a stock would have non-seasonal mortality anyway. For this reason such stocks should be treated with great care.

This is a problem for us in the tutorial, however, since we created a dataset with nicely seasonal growth to demonstrate the upgraded capabilities of LFDA. We would still like to be able to get some idea of the mortality rate for our stock. What we will do, then, is to choose the "best" set of parameters out of our non-seasonal von Bertalanffy model estimates. Look again at the histogram plots to see which fits look the best. The fits with second best score function values for ELEFAN and PROJMAT both look reasonable and have parameters which are close to each other and close to the corresponding seasonal parameter estimates. In fact, the ELEFAN ones are very close to the parameters estimated by ELEFAN for the Hoenig model. Accordingly, we will use these growth parameters, i.e. K = 0.841 and L_{∞} = 180.51, at least for a start. You may well wish to try using some of the other growth parameter estimates later for comparison.

Three methods of estimating mortality are listed under the **Functions** menu. These are the Catch Curve method, the Beverton-Holt method and the Powell-Wetherall method. For details of each of these, see the Technical Appendix.

5.1.5.1 The catch curve method

This method produces estimates of total mortality for each distribution by fitting a regression line through the right-hand side of a length-converted von Bertalanffy catch curve. Select the menu item **Functions | Catch Curve** from the main application window. A dialog box will appear asking you to specify values for K and L_{∞}.

In order to keep similar things together in this tutorial, we have taken you through each of the methods for estimating growth parameters before moving on to estimation of Z. In practice, you may well wish to move directly to estimating of Z immediately after you have used each method for estimating growth parameters. Anticipating this, the package carries over as default values in this dialog box the results of the most recent successful estimation of growth parameters that you have completed. In our case we are using old estimates, so fill in K = 0.841 and L_{∞} = 180.51. The window shown in Figure 5.1.22 will then appear.

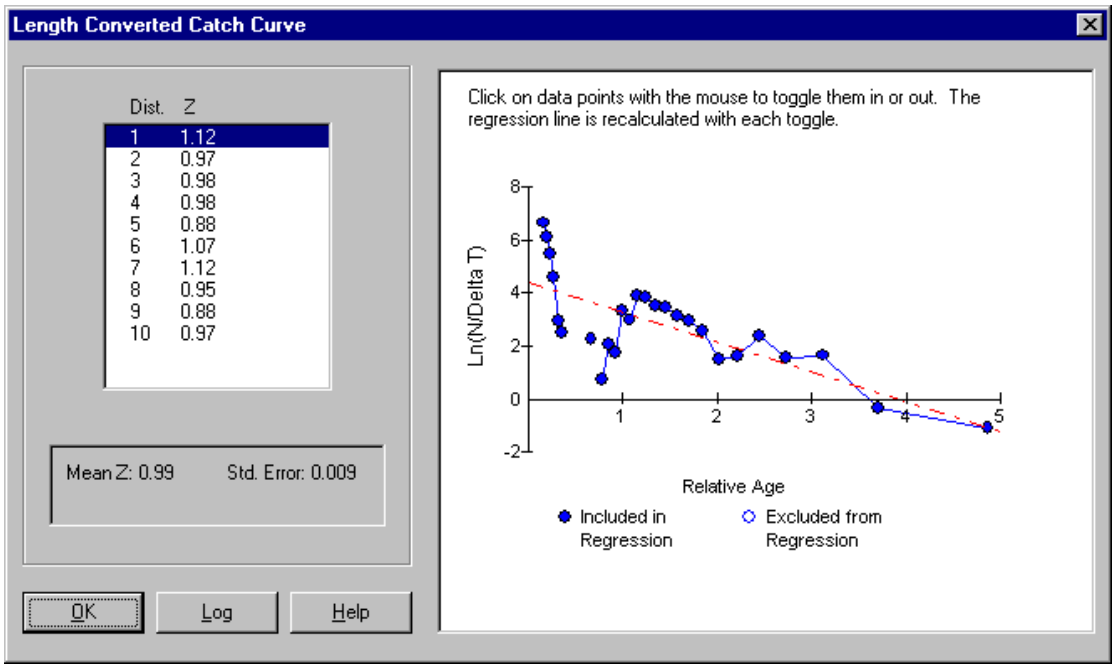


Figure 5.1.22 Converted Catch Curve Mortality Estimate Dialog Box

The list in the left hand side of the window contains an estimate of Z for each of the 10 distributions in our dataset. Beneath this is a panel containing the mean and standard error of these 10 estimates. These estimates are not yet suitable for use as they are based on regressions through all the data points in the catch curve. We really want the regression line to pass through only the descending arm of the catch curve. See Technical Appendix in the Help files for details about the method.

Leave the first of the distributions in the list (1) highlighted. To the right of the list you will see a plot of the points on the catch curve (solid blue circles) as well as the dashed red line that has been fitted through them. You now need to exclude the low age points from the regression, to determine the slope of the right-hand, descending arm of the curve. By clicking on a data point with the left hand mouse button you can toggle points in or out of the regression. Use the mouse to exclude all points in the regression up to and including age 1.1. As you do this you will see the slope of the regressed line (red dash line) changing and the mortality estimate for this distribution (to be found in the distribution list) changes.

When you are satisfied with this fit (in this case $Z = 1.35$ for this distribution) return to the distribution list and highlight the next distribution. This time, toggle off the points up to age about 1.4, just before the highest point in that peak. You should see $Z = 1.40$ in the distribution list. Repeat the process with all the distributions. When you have done them all to return to the list of Z estimates. There is a certain degree of subjectivity in the choice of points to include, but your Z estimates should be close to these:

Distribution	1	2	3	4	5	6	7	8	9	10
Z Estimate	1.35	1.40	1.51	1.45	1.29	1.26	1.31	1.66	1.34	1.54

5.1.5.2 The Beverton-Holt method

The Beverton-Holt method relies on a simple algebraic relationship between the mean length in each sample, the length at first full exploitation, the von Bertalanffy growth parameters and the total mortality rate Z.

Select the menu item Functions | Beverton-Holt *Z* item. A dialog box should appear asking you to specify values for *K*, $L_{¥}$ and *Lc*. *Lc* is defined here as the first length class which is fully exploited. For these simulated data, this is easy to specify, as we did not simulate gear selectivity or movements of the fish or the fishing fleet. Use $L_{¥}$ =180.5, *K*=0.84 and *Lc*=20, which is the lowest length appearing in the simulated data. The window will then extend to appear as shown in Figure 5.1.23.

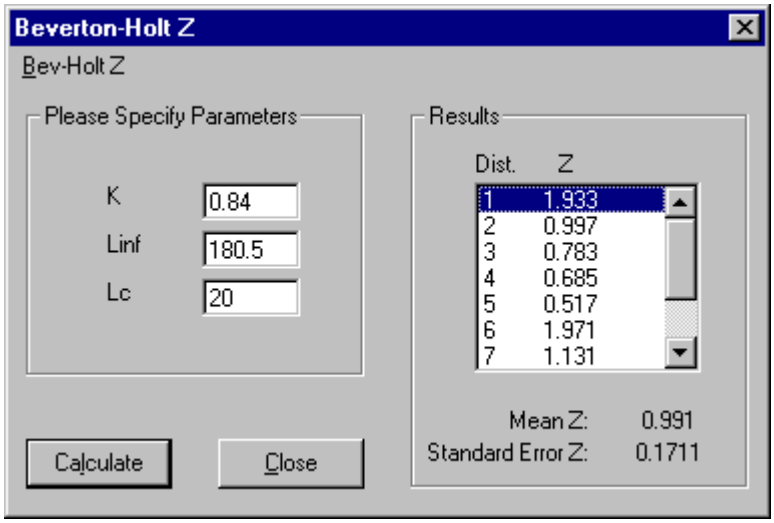


Figure 5.1.23 The Results of a Beverton-Holt Analysis.

One estimate of *Z* is calculated for each separate length frequency distribution as shown in the table below. The mean and standard error of these estimates are also reported, and they are *Z*= 0.991 with Std Error= 0.171. Since you cannot change these estimates at all, as you did in the previous paragraph, they are automatically written to the log file.

Distribution	1	2	3	4	5	6	7	8	9	10
Z Estimate	1.933	0.999	0.783	0.685	0.517	1.971	1.131	0.723	0.650	0.526

5.1.5.3 The Powell-Wetherall method

The Powell-Wetherall method uses a linear regression based on an algebraic relationship between $L_{¥}$, *K*, *Z* and the mean lengths in the length frequencies. However, unlike the other methods, it does not directly calculate an estimate of *Z*; rather it comes up with a series of estimates of $L_{¥}$ and the ratio *K/Z*. While it thus provides yet another means of estimating $L_{¥}$, it is a little more complicated to get estimates of *Z*.

It might seem that all you have to do is to divide an estimate of *K* obtained using one of the other methods for estimating growth parameters by the estimated ratio *K/Z* in order to obtain the corresponding estimate of *Z*. Unfortunately, that is not really appropriate. The reason for this lies in the negative correlation we have mentioned several times between estimates of $L_{¥}$ and *K*. When you change your estimate of $L_{¥}$, the corresponding best estimate of *K* also changes. The best estimate of *K* from, say, the ELEFAN method will have associated with it a different estimate of $L_{¥}$ to that suggested by the Powell-Wetherall method. To be strictly correct, it is necessary, although tedious, to work out for each Powell-Wetherall estimate of $L_{¥}$ a corresponding *K* before calculating a single estimate of *Z*. This is what we strongly recommend you do when analysing one of your own data sets. It is especially important to do this when the Powell-Wetherall estimates of $L_{¥}$ are substantially different from those obtained using the other methods. However, to save you time in this tutorial, we have done those calculations for you.

Select the menu item **Functions | Powell-Wetherall Z**. A window will appear (after a brief period time to perform calculations) that looks quite like the one that you saw while using the catch curve method. This time though, for each distribution an estimate of L_{∞} and *Z/K* is given (Figure 5.1.24). As before, you are given the option to modify each distribution by toggling points in or out of the regression.

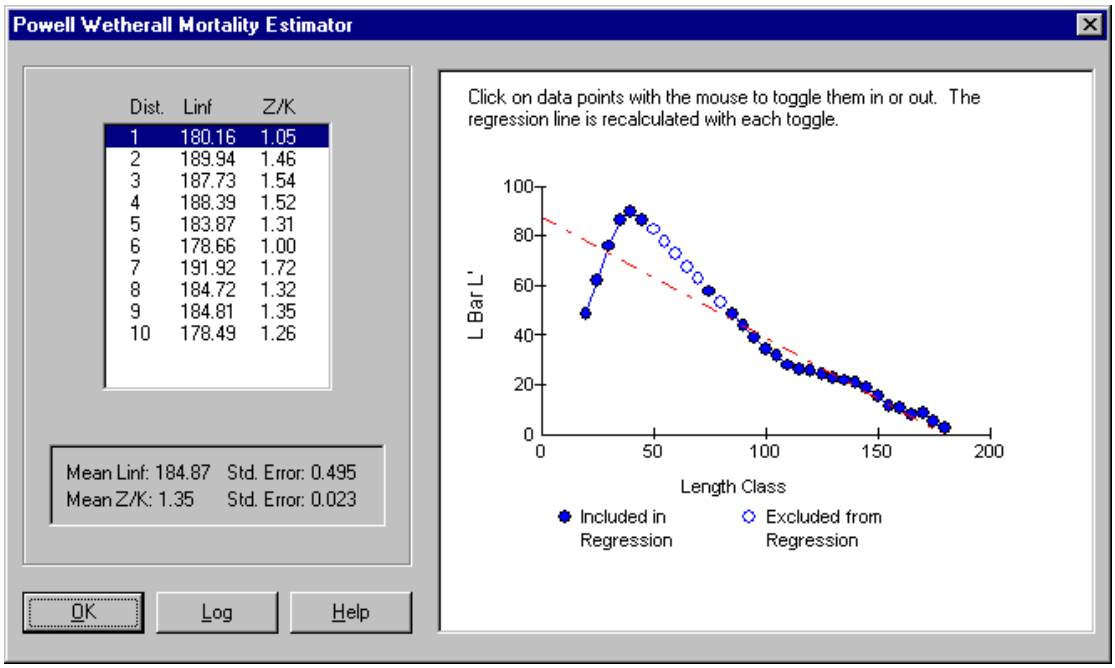


Figure 5.1.24 Powell-Wetherall Method

Highlight the first distribution in the list to produce a plot like Figure 5.1.24. The Powell-Wetherall method calculates, for each length sample in a distribution, the smallest length in the length class *L'* and also *Lbar*, the mean length of all the fish in the whole distribution that are bigger than *L'*. The plot that you see shows a linear regression of numbers in each length class against *Lbar - L'*. The intercept on the x-axis is an estimate of L_{∞}. While the simple theory of the Powell-Wetherall method suggests that the points on this plot should all lie on a straight line, in practice they almost never come close to that over the full range of length classes. However, they often tend to lie much closer to a straight line for larger lengths near L_{∞}. From this plot, it looks as if the points for lengths greater than or equal to 120 lie pretty close to a straight line.

The screen display indicates that all the points that had non-zero numbers of fish in them (i.e. length classes that did appear in the sample) are presently included in the regression. What we need to do now is to remove all points from length classes with a minimum length below 120 from the regression. As before, use the left hand mouse button to toggle points on or off. Toggle off all points corresponding to lengths less than 120. When you are finished, the highlighted estimate in the distribution list should read *Z/K*=1.55 and L_{∞} = 187.84.

This deals with the first length frequency distribution. Now repeat the process of toggling out all points with lengths deviating from the straight line from the remaining 9 distributions. For each distribution you will then have estimates of *Z/K* and L_{∞}. Below these you will see a mean and standard error for all the distributions.

All that remains is to convert these to estimates of L_{∞}, *K* and *Z*. To save time, we have done this for you in the way described earlier. We have calculated the best estimate of *K* for each

Powell-Wetherall estimate of L_{∞}, and then used that to obtain an estimate of *Z*. We used ELEFAN for the estimation of *K* because we used ELEFAN's growth parameter estimates for the other mortality calculations. We fixed L_{∞} at a single value in the grid definition and set the number of grid points for L_{∞} to 1. After the grid had been calculated we maximised over the range (one dimension). We then picked the value of K corresponding to the highest score function. You can check these calculations later at your leisure, if you want. The results are given below.

LF Distribution	L_{∞}	*K* from ELEFAN	*Z/K*	*Z*
1	187.84	0.648	1.55	1.00
2	185.98	0.767	1.56	1.20
3	187.79	0.648	1.83	1.19
4	187.86	0.648	1.66	1.08
5	183.08	0.794	1.17	0.93
6	185.78	0.767	1.51	1.16
7	188.03	0.648	1.80	1.17
8	187.09	0.745	2.00	1.49
9	183.7	0.796	1.36	1.08
10	184.14	0.791	1.90	1.50
Mean *Z* (Std Error)				1.20 (0.050)

5.1.5.4 Comparisons of estimates of Z

So far, we have only calculated estimates of *Z* corresponding to the growth parameters estimated by ELEFAN. On real data we should also try the other parameter sets, especially those produced by the other methods. The Powell-Wetherall estimates were calculated using ELEFAN estimates of *K*, but PROJMAT or SLCA could equally well have been used for that purpose. A normal analysis would involve trying out all sensible combinations.

The true value of *Z* in these simulated data was 1. How well did the methods do in estimating *Z*? The catch curve estimate was a bit high at *Z*=1.4, while the Beverton-Holt estimates varied considerably by distribution, although the mean *Z* estimate was very good at *Z*=0.99. The Powell-Wetherall method slightly overestimates *Z* at 1.20, but with a lower standard error than the Beverton-Holt method. As with the growth parameter estimation, the different methods will perform with varying effectiveness on different datasets. One should in any case not read too much into the present comparison because we are fitting non-seasonal models to seasonal data.

That completes the tutorial. We hope it proved useful and that you now feel ready to try the LFDA package out on your own data. For an explanation of those menu options that we have not gone through here, please consult the Reference section, which also contains some further hints for using LFDA.

All that is left is to exit the LFDA package. To do this, select **Exit** from the **File** menu. You will receive a warning about saving log files (and your data if you have altered it since your last save), which you should definitely take heed of when analysing your own data. If choose not to keep the log file then it will be deleted.

5.2 CEDA Tutorial

The aim of this section is to guide you through the operation of CEDA version 3. If you have not used CEDA before, this tutorial will help you to become familiar with the procedures necessary to use the package. If you have used CEDA version 2 you will notice some changes in version 3, particularly in the design of the user interface.

This tutorial will show you how to use the basic elements of CEDA (i.e. data acquisition, model fitting and projections) by detailed referral to the analysis of two example data sets. By working through this tutorial, you should become familiar with most of the procedures used in CEDA, and should gain sufficient expertise and confidence to continue exploring CEDA's features with your own data. Before continuing with the tutorial it would be useful to familiar with the general principles of model fitting as discussed in the Guide to Fitting Models section of the Help files.

5.2.1 Loading data into CEDA

The Reference and Operating Guide explains the various different formats in which data can be imported into CEDA. There are three available procedures for loading data into CEDA. These include

1. Importing ASCII files
2. Opening CEDA Files. (CEDA 2.0 (*.cd2), CEDA 3.0 (*.cd3).)
3. Manual data entry using the keyboard.

Note: CEDA 3.0 will not open CEDA version 1 files.

5.2.1.1 Importing ASCII files

SQUID.TXT is an ASCII text file of catch and effort data for a squid fishery taken from Rosenberg et al (1990). A printout of the file looks like this:

Illex argentinus 1989

week	totcat	effort	meanwt	n
11.00	5859.29	3380.75	.000354	1
12.00	22185.05	9595.82	.000394	2
13.00	23994.31	9292.13	.000436	3
14.00	24490.95	7803.33	.000481	4
15.00	24904.04	8868.42	.000528	5
16.00	14024.95	9225.93	.000582	6
17.00	12099.11	9520.48	.000636	7
18.00	16630.94	9133.92	.000695	8
19.00	17395.95	10014.95	.000756	9
20.00	8839.38	10147.58	.000821	10
21.00	9944.00	9075.92	.000888	11
22.00	8006.33	8359.62	.000963	12
23.00	3270.66	3982.62	.001000	13
24.00	1980.80	3627.17	.001000	14
25.00	1.80	33.00	.001000	15
26.00	4.04	16.00	.001000	16

To import these data into CEDA you should select the **File | Open** option from the menu. You will then see an open file dialog box. Select Files of type Text (*.txt) from the pull down menu. If you are in the installation directory the file [SQUID.TXT] should appear in the list, otherwise browse to the folder chosen during installation. Open this file and you will then be presented with the ASCII Import Wizard. Import the ASCII data column by column, following the instructions on the wizard (see Importing Data from an ASCII File). The list below shows the ASCII columns and the associate CEDA columns to which they should be assigned;

ASCII Column	**CEDA Column**
week	Timing
totcat	Tot. Catch Wt
effort	Effort
meanwt	Mean Wt
n	(None)

CEDA loads the data and presents it within the CEDA Data Sheet. You will then be asked to specify a filename and location for the newly imported data. Accept the default location and filename of squid.cd3.

Within the datasheet you will notice that 3 additional columns of data have been calculated by CEDA. These columns are Tot. Catch Wt, Catch Wt and Catch Nums. Where these columns come from and what they are used for is explained below and also in the Reference and Operating Guide. The dataset description in the ASCII file is imported into the CEDA dataset as the **File Description** and will appear in the title bar of the main application after the name and location of the current file. The File Description can be changed by selecting **File Description** from the **Edit** menu.

At this stage it is not possible to edit the data. Editing of data can only be done from within Edit Mode. DO NOT enter edit mode at this stage.

Filling Out of Columns

The names at the top of the columns in the CEDA data sheet correspond to the columns you selected from the pull down list when importing with the ASCII Import Wizard. You will notice that the column Tot. Catch Nums, which stands for total catch in numbers of animals, has some numbers in it, even though you have not imported any. This is because CEDA automatically fills out data columns for which it has the necessary information. In this case you have imported the total catch in weight (Tot. Catch Wt) and the mean weight of each animal (Mean Wt). CEDA is therefore able to calculate the total catch in numbers (Tot. Catch Nums) of animals as follows:

$$Tot.\,Catch\ Nums = \frac{Tot.\,Catch\ Wt}{Mean\,Wt.}.$$

The following columns of data can now be seen in the CEDA data sheet.

Tot. Catch Wt	Total weight of all removals.
Tot Catch Nums	Total number of all removals.
Mean Wt	Mean weight of an individual.
Effort	Fishing effort.
Catch Wt	Catch in weight resulting from Effort.
Catch Nums	Catch in numbers resulting from Effort.

(see Filling Out Columns in Edit Mode for additional explanation of the data types).

Decimal Places

You will also notice that the column under mean weight (Mean Wt) shows all zeros. This is because the default number of decimal places set by CEDA is zero. If you go back and look at the print out of SQUID.TXT you will notice that the numbers under the heading meanwt are very small. This is because the units of weight are tonnes and the highest mean weight for an individual squid is about 1 kg. The mean weights in the data window can be displayed properly by changing the number of decimal places in the column. To do this you must move click with the mouse to a position in the column Mean Wt and then select the **Edit | Number** option from the menu, using the mouse or the keyboard as before. Change the number of decimal places to 6 and the number format to Fixed (See Number Formats).

Units of Weight

Note that CEDA expects that all of the weights entered in one data file to be expressed in the same units of measure. If you enter total catch weights and mean weights in different units, CEDA will calculate incorrect numbers of animals. Similarly if you enter numbers of animals and mean weights, CEDA automatically calculates the total catch in weight. The units of the catch weight calculated by CEDA will be the same as the mean weights you entered.

Tip - Standardise the units of weight in your input data files

You have now completed the procedure for importing a text file into CEDA and CEDA has saved your imported data in the file SQUID.CD3, as specified by you at the start of the import procedure.

5.2.1.2 Entering data manually from the keyboard

The editing facility in CEDA means that it is also possible to enter your dataset directly into the software package using the keyboard. Datasets used for analysis using CEDA are usually not very big. There may be considerable amounts of raw data involved, but these are commonly distilled into an unbiased and reliable series of catch and effort data that can be analysed using CEDA. It should therefore not take very long to use this method of data entry. It is particularly useful if the only copy of the dataset you wish to analyse is on paper. Once you have entered the data there is a menu option in CEDA for exporting the data as an ASCII text file for import into other packages (**File | Text Export**).

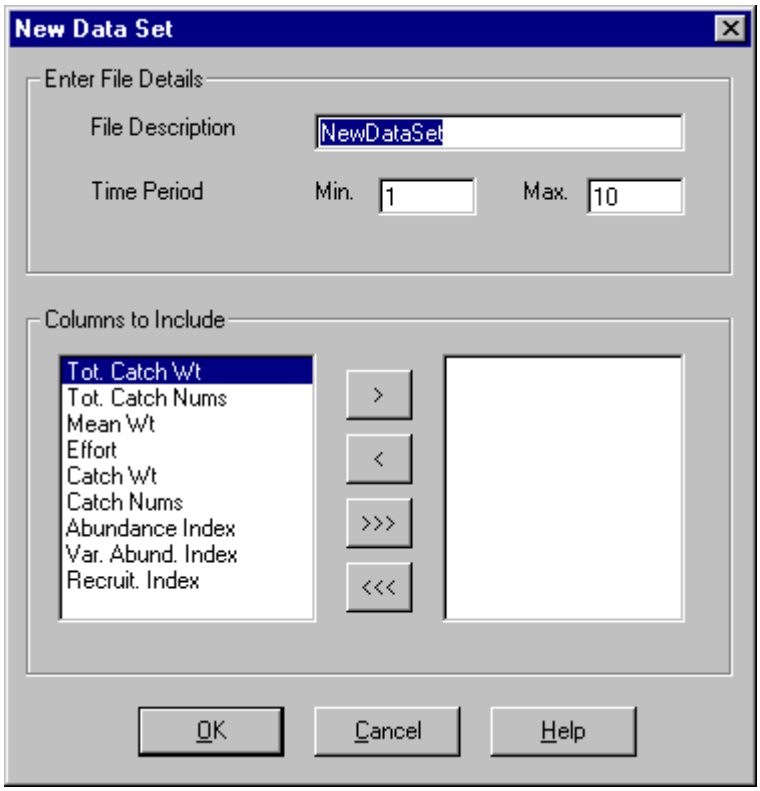
NOTE

The procedure for manual data entry which we suggest you follow here is for demonstration purposes only. The data analysis in the remainder of the tutorial is carried out only using the data you have imported from SQUID.TXT and TUNA.CD2. If you want to save time you can skip straight to the data analysis section. You can return to this section later, when you want to learn how to enter data from the keyboard.

Manual Data Entry

Select the **File | New** option from the menu to start the procedure for manual data entry. You will then see a dialog box which is headed New Data Set (see Figure 5.2.1). You must provide a file description (maximum 40 characters) for the new dataset, for which CEDA provides the default option “New Data Set”. If you want to you can change the file description for something more descriptive, such as 'Squid data entered by keyboard'. If you want to change the description later you can do this by selecting the **Edit | File Description** option from the menu.

Figure 5.2.1 The New Data Set dialog Box



Below the File Description will see text boxes in which you must specify the time period for the dataset. Refer back to the printout of SQUID.TXT. You will see that the time period - i.e. the column week starts from 11 and runs in equal increments up to 26. You should therefore enter 11 in the first box and 26 in the second. CEDA assumes that the time period is divided up into equal increments of 1. Use the mouse or <TAB> to move between the boxes.

You are then presented with a MultiPick box. This allows you to select the target columns into which you are going to type the data. The column headings you choose will be used to construct the data sheet into which you will type the data in edit mode.

The nine available column headings are listed on the left hand side. When column headings are chosen they move to the list on the right hand side. To select a column heading you simply position the highlight over the appropriate title (e.g. Tot. Catch Wt) and then click on the button marked [>]. This moves the chosen heading from the list on the left to the list on the right. If you move a heading, but later decide that you don't want it, you can move it back again by highlighting it and using the [<] button. Using the [>>>] and [<<<] buttons moves the entire list one way or the other.

You will remember that SQUID.TXT contains the columns week, totcat, effort, meanwt and n. You will be entering data from the columns totcat, effort and meanwt using the keyboard (week will be generated automatically for the timing column - see below). You should therefore choose the columns Tot. Catch Wt, Effort and Mean Wt, moving them from the left side to the right side of the MultiPick box.

After you have done this click OK and CEDA will present you with a dialogue box in which you should provide a name and location for the new CEDA file. The file will be given the default extension *.cd3.

You are then presented with the CEDA data sheet which has the timing already entered in the left most column and -1 (CEDA uses -1 as a 'missing value') in the three columns Tot. Catch Wt, Effort and Mean Wt. You can now enter the data from SQUID.TXT in exactly the same way you would in a spreadsheet. Simply type in the numbers in the appropriate place and the arrow keys move the cursor from cell to cell. Start with the cursor in the top left cell. Type 5859 <↵> 22185 <↵> 23994 <↵> etc.

Note that the data sheet is shaded light blue and that **Edit** and **Help** are the only menu choices available. This is because you are now in edit mode, see Edit Mode for a detailed explanation. When you have finished entering the data and you are sure there are no mistakes (although mistakes can be rectified later by returning to edit mode), click on **Edit** menu item. You then have the option to do various things, including inserting or deleting columns (**Insert Column** or **Delete Column**) and changing the timing intervals (**Add Row** and **Delete Row**). When you are satisfied with the data you have entered click on **Edit | Edit Mode** to exit edit mode. CEDA will now fill in columns where appropriate (see Edit Mode). This is the same as before, when CEDA calculated catch in numbers from the catch weight and mean weight data which you imported directly from SQUID.TXT.

One final thing you must do upon leaving edit mode is include all data points for analysis. This is done by clicking the check boxes for each time interval in the column marked Include. The data you have entered by hand are now ready for analysis using CEDA.

5.2.2 Squid tutorial

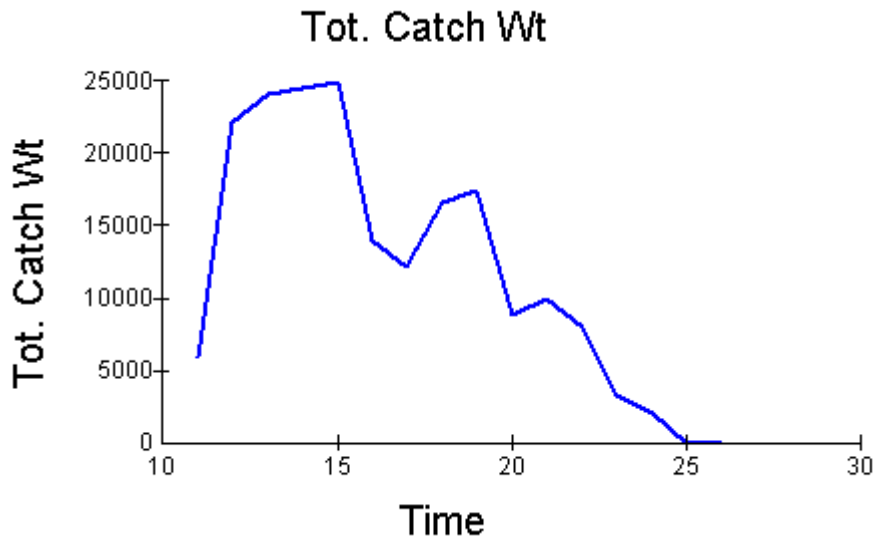
The analysis of the squid data will be done using the data which has been imported previously from the ASCII text file SQUID.TXT (If this has not been done, see Importing ASCII Files). Select the **File | Open** option from the menu. (At this stage, if you have a file open or have been manually entering a file using the keyboard, you will be prompted to save the loaded file). Load the file Squid.cd3

The squid data will appear in the CEDA data sheet. Prior to starting on the data analysis we suggest you spend a few minutes familiarising yourself with the options for data presentation. Before starting a detailed analysis it is always worth taking a step back and looking at the data. The graphing facilities under the menu option **Data | Data Plots** are useful for this purpose (This is covered in the first stage of the tutorial below). You can plot any of the columns in the dataset against the timing column to look for trends in the data. You can also print out the dataset using the **File | Print** option.

5.2.2.1 Selecting model options

If you select the **Data | Data Plots** option and plot Tot. Catch Wt against Time you will see Figure 5.2.2 below.

Figure 5.2.2 Total Catch Plotted Against Time for the Squid Dataset



This shows a rapid increase in catch over the first week (11 to 12), which levels off up to week 15 before a gradual decline to zero at week 25, with a few humps and bumps on the way. If you plot catch in numbers Tot. Catch Nums against time you will see an increase in catches for the first week which is followed by a steady fall. The No Recruitment section explains that the most suitable method of analysis for this type of data from a squid fishery is a depletion model with no recruitment. This method of analysis is available in CEDA. You are now ready to analyse the data.

The options for data analysis are found under the menu headings Fit and Projections. To specify the analysis method you want to use select Fit | New Fit from the menu. You will notice that none of the other options under the menu heading Fit are available apart from Fit Manager. This is because they all refer to analyses which are done after the initial parameter estimation. Once the initial parameter estimation has been done all of these options will become available. When you select the Fit | New Fit option, you are presented with 5 available models in a dialog box. Select No Recruitment. You are then presented with a dialog box in which you must specify a number of input parameters (see Figure 5.2.3 below). The dialogue box contains a mixture of text boxes and check boxes. Those options which are not applicable for the current model and error model with be 'grayed' out. For the No Recruitment model you are first prompted to enter a natural mortality rate *M*.

Figure 5.2.3 Model Input Parameter dialog box for the No Recruitment model.

The units of the natural mortality rate (e.g. per day, per week or per year) should be the same as the time period of the data. The time period of the squid data is one week. The natural mortality rate should therefore be expressed in terms of weeks. A typical weekly natural mortality rate for squid is 0.05. You should enter 0.05 at the prompt.

You then have to specify which of the three available error models (Least Squares (Unweighted), Log Transform or Gamma) you want to use when fitting the data. All three error models will be tested in the course of this tutorial. The first one to try is the fastest and simplest, Least Squares (Unweighted). To select this option click on it using the mouse. Alternatively use the <TAB> key to move to the Error Model box and then use the up and down arrow keys to highlight your selection.

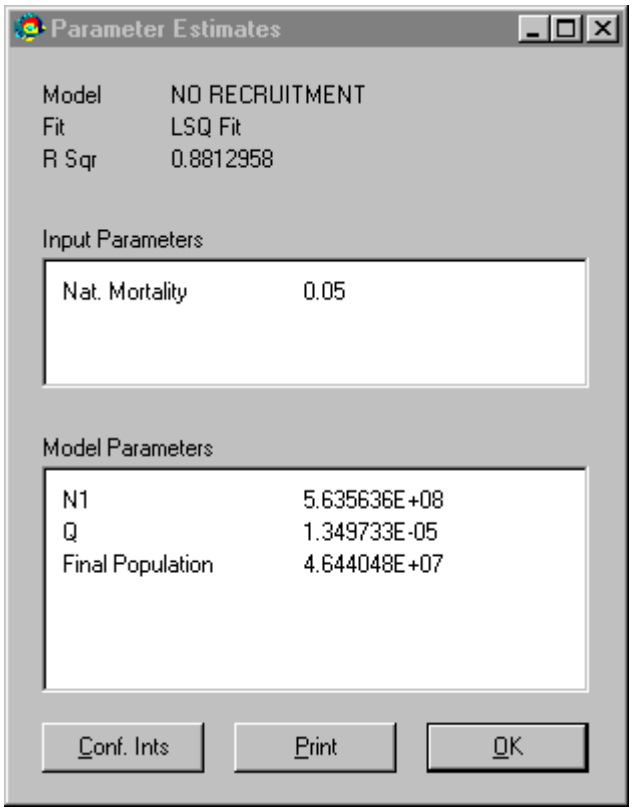
You will also notice two further grayed out options Set Starting Estimates Manually and View Minimisation Graphically, both of which are unavailable for the combination of No Recruitment and Least Squares (Unweighted).

Once you are satisfied that you have made the correct choices in the dialog box, close it by clicking on the OK button.

5.2.2.2 Initial parameter estimates

Once you have confirmed the choice of input parameters, CEDA takes a small period of time (this may be a few seconds on slower computers) to fit the model to the data. After the model has been fitted, CEDA displays two new windows called Parameters Estimates (Figure 5.2.4) and Expected and Observed Catch (Figure 5.2.5).

Figure 5.2.4 The Parameter Estimates Window



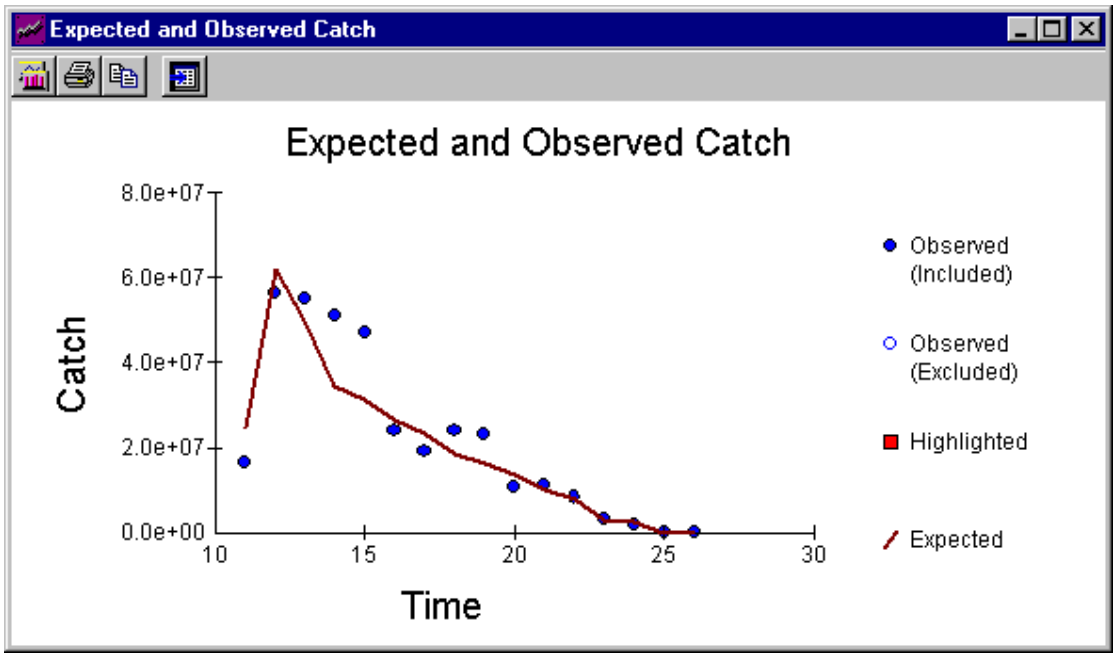
At the top of the Parameter Estimates window is the population model that has been fitted (in this case No Recruitment was selected), the error model selected (Least Squares) and a goodness-of-fit measure (R^2).

Below this is a white box labelled Input Parameters, which contains a list of the user input parameters for the current fit. In the case of the No Recruitment model the only user input parameter is natural mortality *M*.

A second white box labelled Model Parameters contains the estimated parameters calculated by CEDA. For the current Population Model and Error Model there are 3 estimated parameters:

N1	Initial population size in Numbers
q	Catchability coefficient
Final Population	Final Population Size in Numbers

Figure 5.2.5 Expected and Observed Catch. For more information on this type of graph see Graphs of Expected and Observed Catch / CPUE



The graph shows a reasonable match between observed and expected catch, but that does not necessarily mean that the fit is adequate. In order to assess how well the model fits the data it is necessary to consider a number of other outputs provided by CEDA, collectively known as the diagnostics. The first of the diagnostics you should look at are the residual plots, which are discussed further in the Residual Plots section.

5.2.2.3 Examining the fit

If you browse around the CEDA data sheet in the main application window you will notice that the results of your first run have been added to the dataset. The following columns of data have been added:

Exp CPUE	Expected (estimated) CPUE.
Obs CPUE	Observed CPUE.
Exp Catch Nums	Expected catch in numbers.
Pop Wt	Population abundance in weight.
Pop Nums	Population abundance in numbers.

The residual plots are found under the Graph menu option. Select this option and open the following graphs by selecting them in turn from the Graph menu.

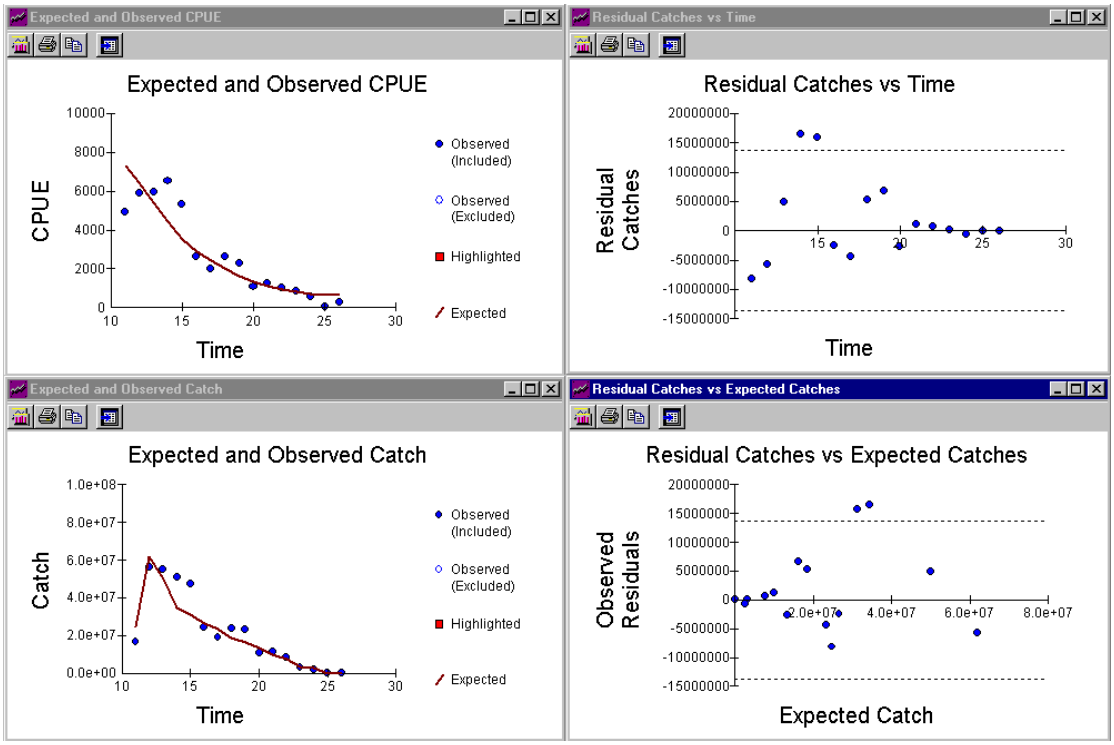
Residual Catches vs Time
Residual Catches vs Expected Catch
Expected & Observed CPUE

You will now have a total of four graph windows loaded on the screen (including the Expected and Observed Catch graph window produced after running the model). Organise these four windows so you can see all four graphs.

Note: You may find it useful to minimise the Parameter Estimates window and the main CEDA application window to create more space on the desktop.

These are diagnostic graphs, giving you information on how well the model fits the data. Your screen should contain the four graphs shown in a layout similar to that shown in Figure 5.2.6.

Figure 5.2.6 Diagnostic Graphs



Examine the two residual plots. If you are unsure what to look for, refer back to discussions in Residual Plots. Briefly, in a "good" residual plot, the points should be evenly scattered in a horizontal band. There should be no sign of trends, curvature or tendency for scatter to increase or decrease as you look along the horizontal axis; nor should there be any appearance of long sequences of points consistently on one side or the other of the horizontal axis.

In both of the plots shown here, there is a tendency for the average absolute values of the residuals to change along the horizontal axis, giving a "triangular" appearance; as you move to the right, there is a decrease in absolute value in the time plot, and an increase in the expected value plot. Do not be misled by the seemingly "high" R^2 value; the evidence of these two residual plots shows that the fit is in fact poor.

What other conclusions can be drawn from these diagnostic graphs? Under the assumption of no recruitment, population numbers should systematically decline due to fishing and natural mortality. This should be reflected by a consistently decreasing trend in CPUE (although there may of course be some random fluctuation). However, this dataset does not show such a trend. Look at the graph of Expected & Observed CPUE; over the first four points, the observed CPUE actually **increases**. More background information is needed to diagnose the problem.

The data in SQUID.TXT are for a fishery targeting the squid species *Illex argentinus*. This species is known to migrate into the area of fishing at about the same time as the fishing season and the collection of catch and effort data begin. If there is significant migration into the area after the start of the fishery, the basic assumption of no recruitment is violated. In this case other sources of information (e.g. data from sampling by scientific observers) provide evidence of migration into the area for the first three weeks of the fishery. The commercial CPUE at this time increases, due to immigration, and does not tell us anything about the total stock size. It is therefore sensible to re-do the data analysis without the first three CPUE data points. Note that it is vital that we continue to use all of the total catch data. Without this the cumulative catch, and therefore the stock size, will be underestimated (see Data Requirements).

Saving the Fit

Before proceeding to re-do the fit you should save the working you have done so far. This is done by adding the fit to the Fit Manager. Each fit is logged separately within the CEDA file. Unlike CEDA 2.0, with CEDA 3.0 after a fit has been added to or deleted from the fit manager, the file, including all fits is saved to disk. The CEDA data file with which you are working (SQUID.CD3 in this case) will get bigger. This is convenient because it means that a record of your analysis is saved with the data.

Select **Fit | Fit Manager** from the menu. This brings up the fit manager, a tool which allows you to save, delete and reload different models you have run on your dataset. We are going to add the current model to the fit manager. Press the **Add Current Fit** button. You are then prompted to enter a description to identify your work. This is important, because later you will need to identify each fit you have logged when you want to retrieve it. We suggest you call the first logged fit *Illex1*. You should type this in the description box and click on the OK button. After a short delay whilst the details are written to disk, the fit you have just saved will appear within the grid on the Fit Manager. You can reload a logged fit and the associated plots later on. This is useful for making comparisons between different fits (See Saving, Loading and Deleting Fits for detailed information on using the Fit Manager). Dismiss the fit manager by pressing the Close button.

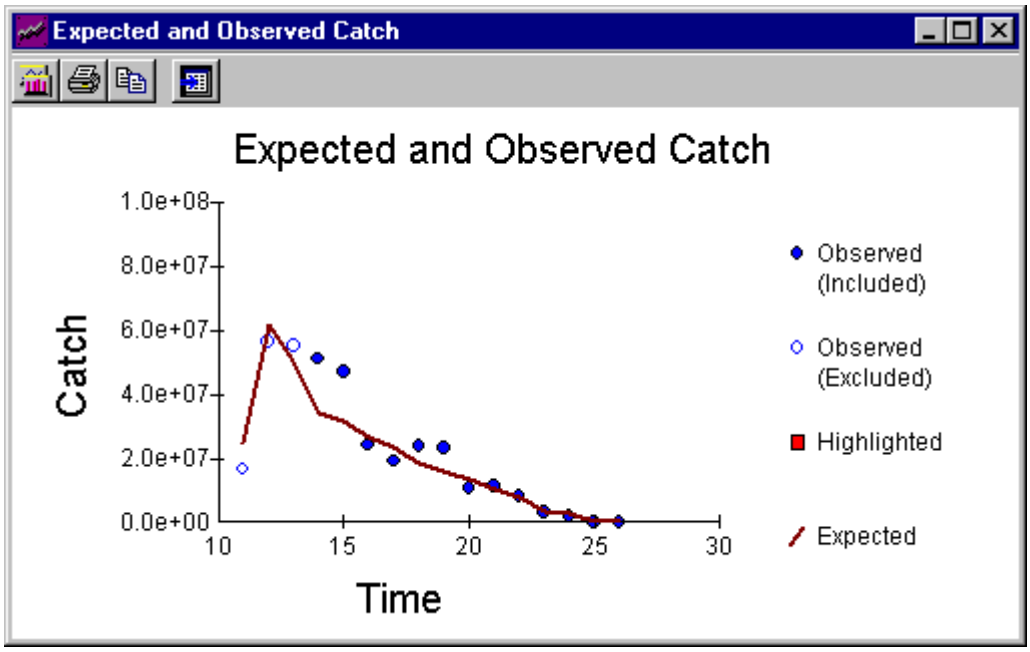
Re-Doing the Fit

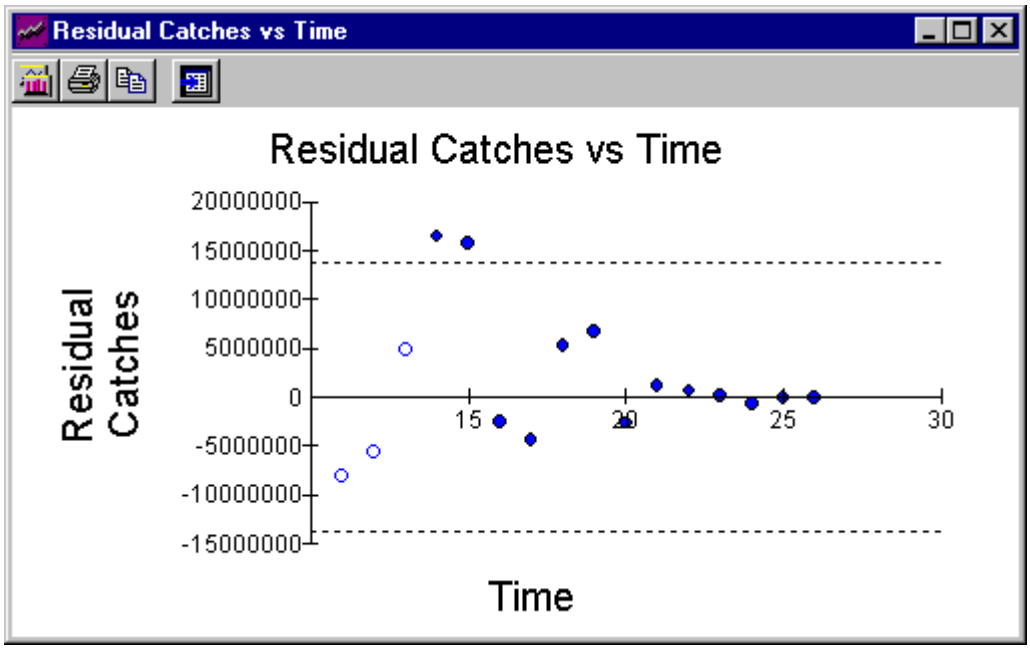
Now we can address the problem with the catch data at the start of dataset. It is possible to temporarily remove ('toggle out') some of the data points from the analysis by using the mouse to toggle points directly on the 4 diagnostic graphs which are currently maximised. This a useful feature in CEDA, but it should be used very carefully. You must always have a good reason for leaving out data from your analysis. Data should not be left out just because they do not appear to fit in with expected results. Other information, such as that provided above for the *Illex argentinus* fishery, is required. If you leave out data without good reason, even though your results look good, they might be dangerously misleading (see Guide to Fitting Models).

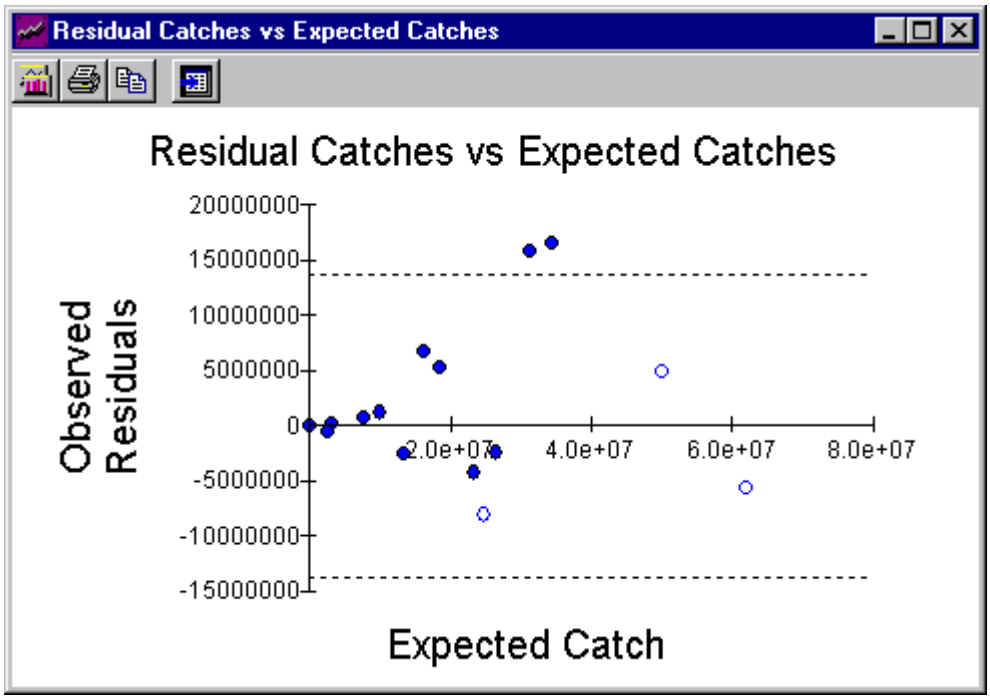
CAUTION: Never omit data points from the analysis unless you have a good reason to do so.

We shall now temporarily exclude the three offending points from the analysis. Select the graph of Expected and Observed catch and with the mouse click on the first data point in the time series (week 11). A message box will appear asking if you wish to continue and clear the current fit. Select ‘Yes’ . The data point which you just clicked has turned from a solid circle to an unfilled circle which represents a point that will not be included in analysis, i.e. it is toggled out. Notice that the point has been toggled out in all the other plots and also from the CEDA data sheet in the main application window. Returning to the graph of Expected and Observed Catch, use the mouse to toggle out the following two data points so that the first three points are all excluded. You will see the display below:

Figure 5.2.7 The Squid Dataset with Three Points Toggled Out.



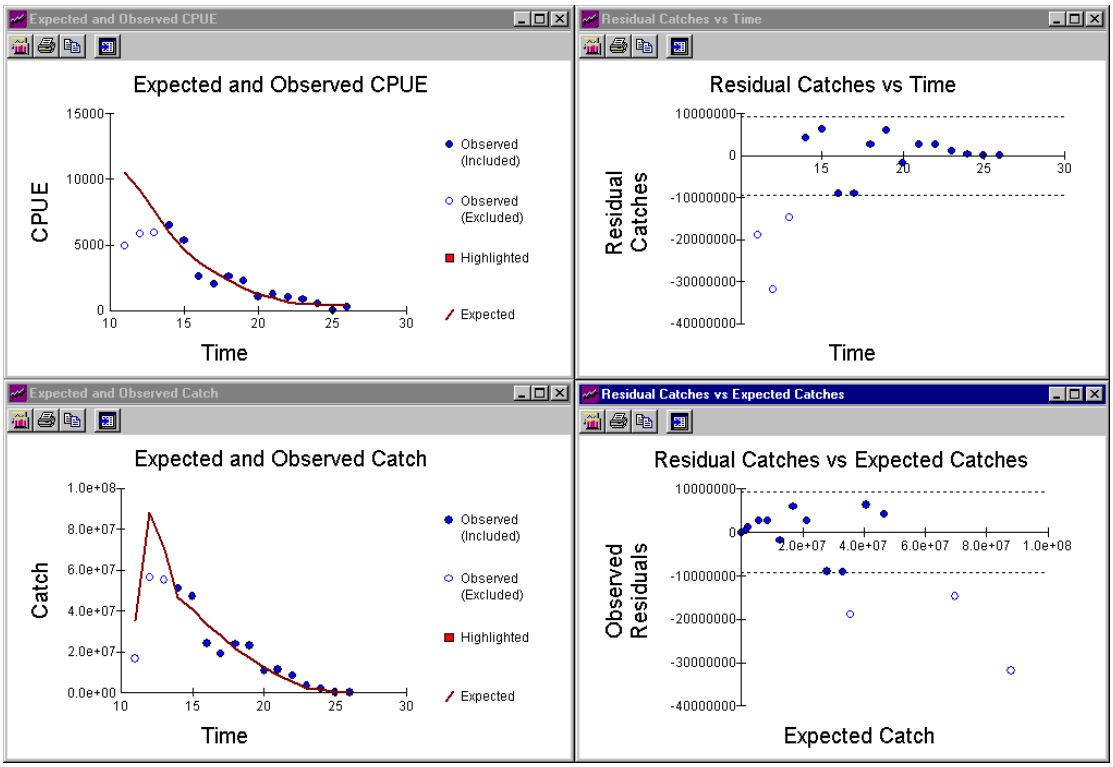




When working with the diagnostic graphs it may be useful to know where a data point on the Observed and Expected Catch graph is to be found on the residual plots. Therefore CEDA has provided the ability to highlight a data point on all graphs, thereby allowing the user to decide whether or not he wishes to toggle out the point. This is particularly useful for the plot of Residual Catch vs Expected Catches which does not share the same X-axis. For example, return to the plot of Observed and Expected Catch and click the right hand mouse button over the first data point. That point and the same data point in all the other plots will change to a red square. Turn off the highlighter by clicking once more with the right hand mouse button over any of the highlighted points.

From the menu select Fit | New Input Parameters. CEDA will clear all graphs and display the form for input of the model parameters. Input parameters and model selections will be the same as those for the last fit used on the data set. CEDA will then allow you to select new input parameters if you want to. For this example you should leave the inputs the same (M = 0.05, Least Squares (Unweighted) error model). When you click on OK, CEDA will refit the model to the remaining data points. You will again see the Parameter Estimates window and a graph of Expected & Observed Catch. You will see immediately that the line of expected CPUE fits the data points much better than in the previous fit. Return to the main window and using the Graphs menu once again bring up the same four diagnostic graphs. This will give the display in Figure 5.2.8.

Figure 5.2.8 Diagnostic Graphs After Removing the First Three Points



The points in the residual plots are scattered in a more horizontal band than they were before. Leaving out the three suspect Catch data points has removed some of the triangular pattern. However, there now appear to be two other potential outliers - points which fall on or outside the horizontal dotted lines on the residual plots (remember that only the included data points count now - the open circles have been eliminated).

The presence of these potential outliers is of concern. First we will compare this fit with *Illex1* and then we will address the problem of the potential outliers.

Comparing Fits

You can compare these new plots with those you obtained previously when you were using the whole dataset. Dismiss the plots from the screen and select **Fit | Fit Manager** to bring up the fit manager. Press **Save Current Fit** and save the current fit as *Illex2*. Using the mouse to highlight the previous fit we saved, called *Illex1* and press the **Load Fit** button or double click with the mouse on the appropriate fit in the fit manager. You are asked to confirm that it is OK to clear the existing fit. Providing you have logged the existing fit then it is OK, because you can load it up again whenever you want to. Click on OK. This brings back the Parameter Estimates window and plot of Expected & Observed Catch that was displayed after fitting. Using the **Graph** menu display the residual plots also.

The exclusion of the first three points also changes the parameter estimates. Using the entire dataset (*Illex1*), the estimate of ***N1*** is 5.636E+08 compared with 4.969E+08 using the reduced dataset (*Illex2*); the estimates of ***q*** are 1.349E-05 and 2.204E-05 respectively.

Is *Illex2* the best fit that can be obtained? Not necessarily; even though the R^2 is higher (0.916 for *Illex2* and 0.881 for *Illex1*), there is still the problem of the outliers. Consideration should be given to the choice of error model. Is Least Squares (Unweighted) the most appropriate? Thus far it has been used only because it is the fastest to run on the computer. Remember there are two other error models which could be used: Log Transform and Gamma.

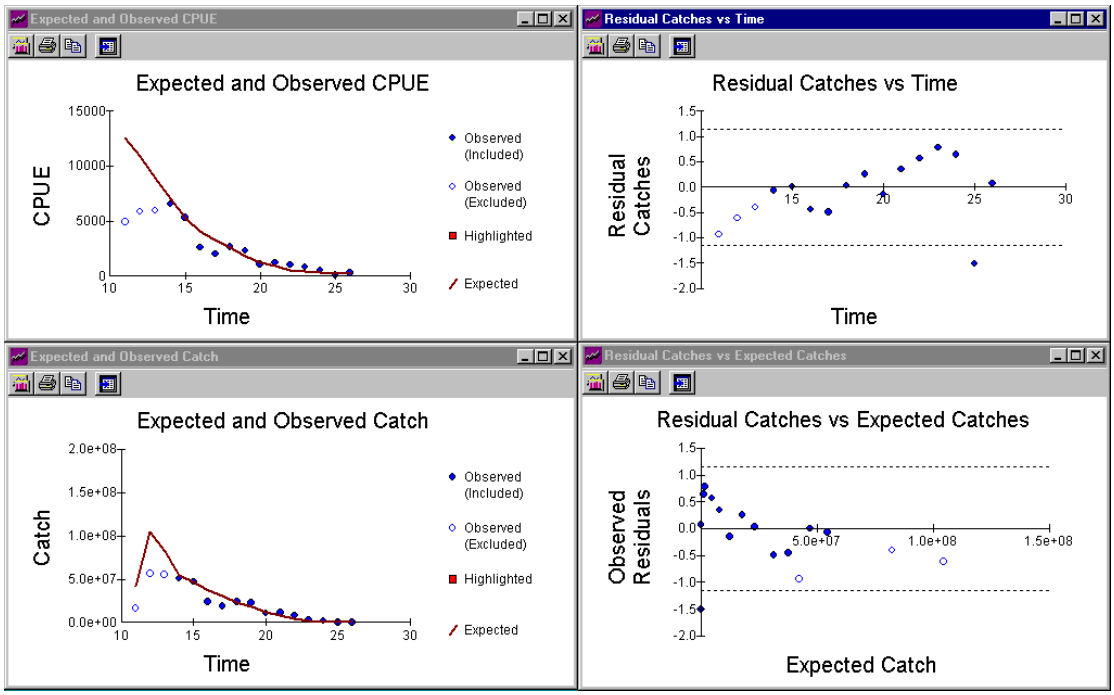
Using Different Error Models

We will now repeat the *Illex2* fit using the other error models to see what difference it makes. For the same reasons as before, the first three data points will be excluded. If the first three data points are not toggled out you should toggle them out now. Select the option **Fit | New Parameters** from the menu (note that you can only select this if you have a fit already loaded). Providing *Illex2* was the last fit you had loaded, the first three data points will already be toggled out.

The dialog box for specifying the Model Input Parameters appears on the screen. You should specify an ***M*** of 0.05 as before and select the Log Transform error model. As before you should not opt to set starting estimates manually. If you select the wrong option by mistake, simply click on Cancel and start again.

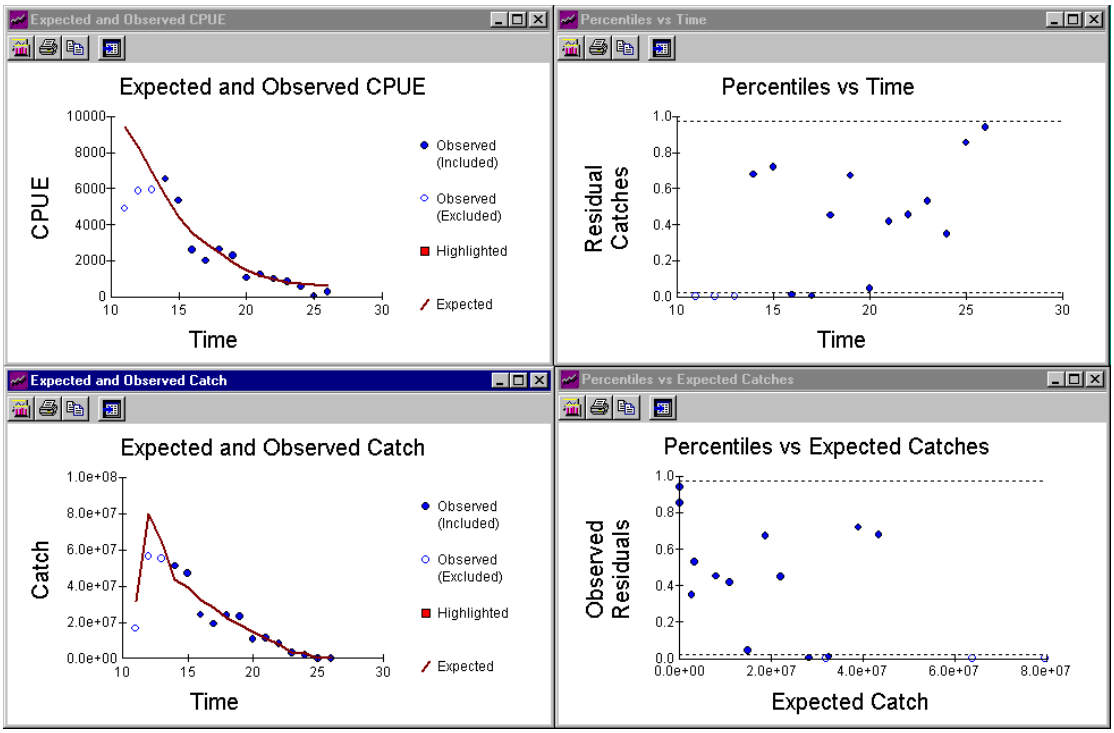
Select the same four diagnostic plots as before (see Figure 5.2.9). You will see obvious trends within the residual plots, and an isolated outlier at week 25. There are no good reasons from either squid biology or the operation of the fishery to exclude the data point at week 25 (see Outliers). This is probably the best fit we are going to achieve using the log transform error model. The strong trends in the residual plots are sufficient evidence to conclude that this error model is not appropriate for this dataset.

Figure 5.2.9 Diagnostics for Log Transform Error Model



Save this fit using the fit manager (calling it *Illex3*) and repeat the same process with the gamma error model. Remember that the first three data points should be excluded. With the gamma error model, residual plots are replaced by percentile plots (see Residual Plots). Use the Graph menu to display the percentile plots and the graphs of Expected & Observed CPUE and Expected & Observed Catch, as usual. You will see the display shown in Figure 5.2.10.

Figure 5.2.10 Diagnostics for Gamma Error Model



Percentile plots can be read very much the same way as residual plots. Outliers are points that are very close to the top (1) or bottom (0) of the graph. It is harder to tell whether an outlier is "extreme" or not with a percentile plot. However, pattern, trend and sequences of points all on one side or other of 0.5 on the vertical axis are still the things to look out for. The correct choice of error model may not solve all of these problems, but it should help with some.

There is at least one data point which could be considered as an outlier, at week 17. Apart from this problem with outliers, which is common to all of the fits we have tried here, the gamma percentile plots do not indicate any other problems with the fit. You will remember that, by contrast, the least squares residual plots showed a triangular pattern, even after the first three points were excluded, and the log transform residual plots showed strong trends. The gamma error model therefore appears to be the most suitable for this dataset.

However, the presence of an outlier is still a problem. There is no particular reason to omit this data point from the analysis and to do so without a good reason would be wrong. Outliers cannot be omitted just because they are outliers - the problem may be with the model rather than the data. One approach when faced with this problem is to test how influential the outlier is on the parameter estimates made by CEDA. This can be done by omitting the outlying data point temporarily and re-running the analysis to see what difference it makes (see the later section 'Re-doing the Fit'). In this case the parameter estimates are not very different with or without the Catch data point at week 17. This is a very different result to the tuna data analysis later in this tutorial.

Since the parameter estimates are relatively insensitive to the presence or absence of the outlier, we shall continue the analysis using all of the data points (except of course the first three points). It is better to do this than to exclude the outlier, because the arbitrary exclusion of the most "extreme" point would lead to underestimation of the variability in the data and to confidence intervals which are artificially narrow.

Use of R^2

The conclusions reached so far on how well the models fit the data have been made without directly using the goodness of fit (***R²***) values. We have concentrated on an examination of diagnostic graphs (residual plots etc.) and some consideration of the effects of known biological phenomena. It would in fact be wrong to use ***R²*** values when deciding which error model to use. The three error models (least squares, log transform and gamma) use different methods of calculating ***R²***. Any comparisons between the ***R²*** values reported by different error models are therefore invalid. It is also wrong to use ***R²*** values as a basis for excluding particular data points. This is because the resulting data series will be of different lengths and their respective ***R²*** values therefore cannot be compared.

However, it is valid to compare values of ***R²*** between fits where data series of the same length and the same type of error model have been used. But this should never be used as the only measure of the validity of the fit of the model, because it tells you nothing about the distribution of the residuals.

WARNING: Never use ***R²*** as the only measure of how well a model fits the data.

Gamma Fitting

When you are analysing your own data, if you also conclude that the gamma error model is the most suitable, ideally you should proceed using only gamma fitting in subsequent runs. The problem is that gamma fitting can be extremely slow, especially when generating confidence intervals. When analysing your own data, this may not be a problem, especially as you should not have to generate confidence intervals very many times; if it has taken 20 years to collect a dataset, then leaving a computer running for half an hour is a small price to pay for an improved analysis! However, the aim of this tutorial is to demonstrate the features of CEDA within a reasonably short time. We therefore suggest that you use the Least Squares (Unweighted) error model for all subsequent work on the squid dataset. The least squares residual plots are not perfect, but they are much better than those for the log transform error model, and the parameter estimates are close to those of the fit using the gamma error model.

You should log the current fit (i.e. the one using the gamma error model), calling it *Illex4*, and proceed to the next part of the tutorial.

Sensitivity to M

The next step in the analysis of the squid data is to investigate the sensitivity of the fit and parameter estimates to different values of the natural mortality rate ***M***. The little that is known about natural mortality rates in comparable squid species suggests that the true value is likely to be within the range 0.01 - 0.10 per week. Try re-fitting the model (use the menu item **Fit | New Input Parameters**) with several values from this range, always using the least squares error model. Remember to log the fits (use names like M0.01 and M0.03) so that you will be able to retrieve them later, and also to make sure that the first three points are toggled out every time. Look at the effects of varying ***M*** on the parameter estimates, residual plots, plots of observed and expected CPUE, and also the plot of observed and expected catch. Also, make a note of the ***R²*** value for each fit. It is valid to compare values of ***R²*** between these fits, because the length of the data series and type of error model are the same.

You should find that varying ***M*** makes little difference to the quality of fit, both in terms of the diagnostic plots and the value of ***R²***. One corollary of this is that it is obviously hopeless to try to use this dataset to estimate ***M***! Unfortunately, varying ***M*** does make a large difference to the parameter estimates; as ***M*** is lowered, the estimate of ***N1*** decreases and the estimate of ***q*** increases. These parameters are therefore described as sensitive to the choice of ***M***.

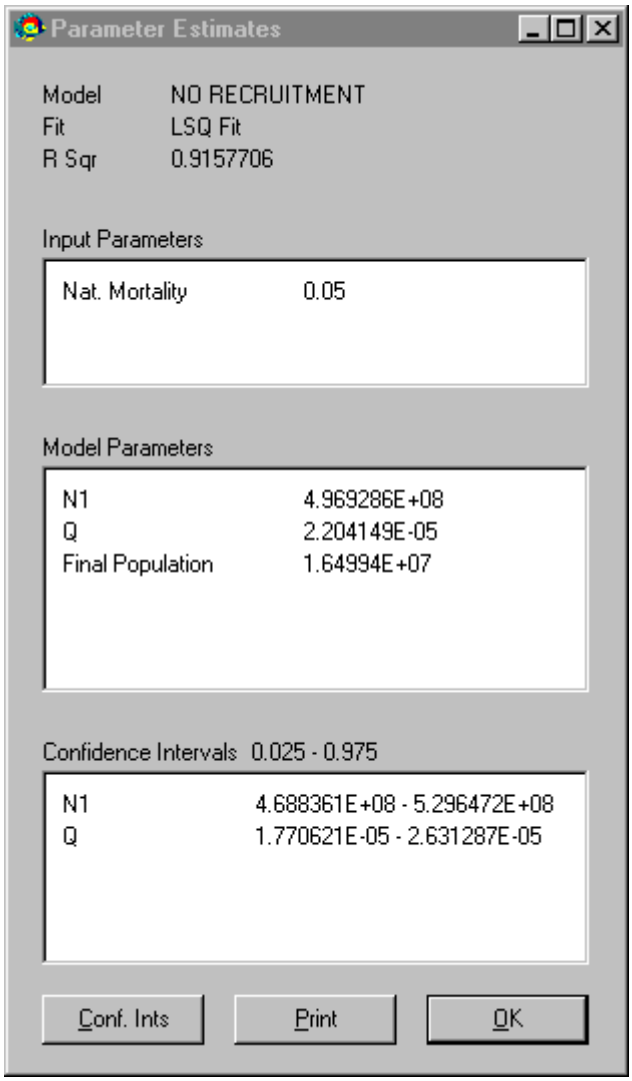
In the absence of better information on the value of ***M***, all that can be done is to present a range of results. This applies both to point estimates of parameters and to the confidence intervals which you are about to generate using CEDA.

5.2.2.4 Generating confidence intervals for the squid data

One of CEDA's most useful features is that it allows you to generate confidence intervals for the estimated parameters. The method used to generate confidence intervals, known as bootstrapping, is explained elsewhere in the manual.

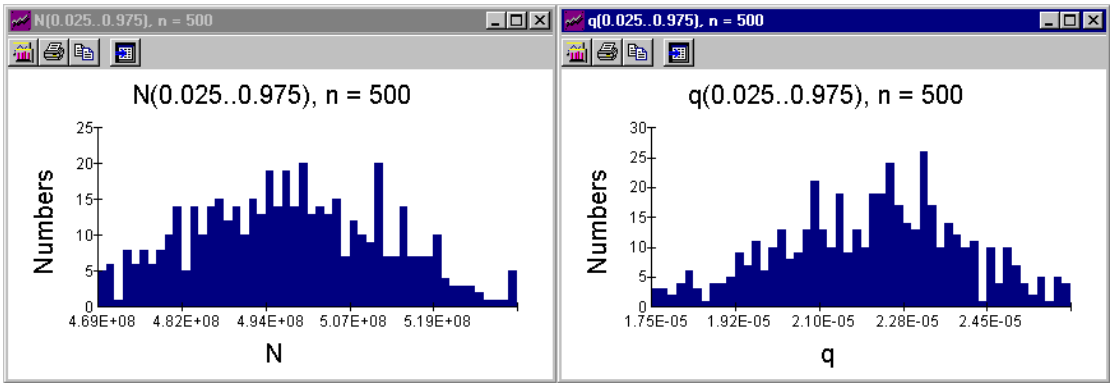
Make sure that the logged fit *Illex2* is currently selected (use Fit Manager for this). Then select the menu option **Fit | Generate Conf. Ints** or press the **Conf. Ints** button on the Parameter Estimates form. A progress bar showing that the bootstraps are being evaluated appears on the screen. The confidence intervals are generated after a few seconds. You are then asked to specify the confidence interval you require. The defaults are set for a two-sided 95% interval; if instead you wanted a lower 90% interval, for example, you would enter 0.1 and 1 in the two cells of the dialog box. When you have entered the confidence intervals you require (we suggest you start with the defaults), click on the OK button. The Parameter Estimates window then resizes itself to display a new white output box which displays the confidence intervals for *q* and *N1* (See Figure 5.2.11).

Figure 5.2.11 The parameter estimates window resized to display Confidence Intervals.



This screen can be printed by clicking on the Print button. The frequency distributions of the bootstrap estimates of *N1* and *q* whose values lie within the selected confidence intervals are now added to the list of plots under the Graph menu. Using the Graph menu open up the two histograms *q* (0.025..0.975), *n*=500 and *N* (0.025..0.975), *n*=500. These plots show the frequency distributions of the parameter estimates (*q* and *N1*) and are illustrated below in Figure 5.2.12.

Figure 5.2.12 Frequency distributions of Least Squares bootstrap parameter estimates



The confidence intervals you specified in the previous dialog box determine how much of these graphs you see. The horizontal axes are adjusted according to the limits on display. The maximum possible extent of the graph would correspond to specifying values of 0 and 1 for the confidence interval in the dialog box titled Enter High and Low Confidence Levels (i.e. a 100% confidence interval).

The reason for not showing the whole graph is that, with some datasets and models, all that is then displayed is a "spike" at one point for ***N1***, and a horizontal axis ranging from 0 to 1E+29. Such graphs are not terribly informative. The reasons why they sometimes occur, and the solution to the problem (using a smaller range in the dialog box), are discussed in the sections discussing Confidence Intervals and Fitting Models.

These frequency plots are intended to give a qualitative impression of the shape of the confidence intervals. In Figure 5.2.12, the distribution of *q* is slightly left-skewed. Sometimes you will see one or other highly skewed plot; if the skew is to the right, this indicates that the upper limit of a two-sided confidence interval will be much further from the point estimate than will the lower limit.

Close down the two frequency plots and save this fit (using Fit Manager) to avoid losing the confidence intervals you have just generated (call this fit *Illex5*). If you want to re-display the frequency click on the **Graph** menu and you will see that CEDA has added them to the list of plots available for display.

Sensitivity to *M*

The confidence intervals you have just calculated are for a natural mortality rate ***M*** of 0.05 per week. However, as mentioned earlier, the parameter estimates are sensitive to ***M*** and there is some uncertainty in the value used here. The true value is probably somewhere between 0.01 and 0.1. Therefore assuming a particular value of ***M*** in the calculation of the confidence intervals may lead to an under-representation of the uncertainty in the parameter estimates. One approach in this situation is to calculate confidence intervals for a range of values of ***M*** and combine these to form an overall conservative confidence interval.

For ***M*** of 0.01, 0.05 and 0.1, a range of 95% confidence intervals for ***N1*** and ***q*** can be calculated, as shown in the following table:

M (per week)	**N1** (10+E08) ***N1*** (1e+8)	***q*** (10E-05) ***q*** (1e-5)
0.01	3.7 - 4.1	2.2 - 3.2
0.05	4.7 - 5.3	1.8 - 2.6
0.1	6.6 - 8.2	1.2 - 2.0

Combining these intervals would give you an interval that you are sure has **at least** a 95% chance of containing the true values. You can do this by selecting the lowest lower limit and the highest upper limit of the 95% CIs from the above table. This gives an overall 95% confidence interval for ***N1*** of 3.7E+8 - 8.1E+8. Do not be concerned if the Confidence Intervals you have generated differ slightly from those shown in the table. This happens because CEDA uses a random seed for bootstrap sampling which will be different each time Confidence Intervals are generated. See Confidence Intervals for a more detailed discussion and some warnings.

5.2.2.5 Making projections

You have made a reasonable assessment of the population parameters of the squid stock (and their associated uncertainty), based on the data provided in SQUID.TXT. Using these parameter estimates, and their estimated uncertainty, CEDA can make projections to investigate the potential effect of possible future levels of catch and effort on the population. This is useful both for making forecasts and for investigating the effects of alternative management strategies. The CEDA population projection facility is accessed under **Projections** on the main menu bar.

Before starting the projection, you should make sure that you have the correct fit loaded. The last one we worked on was *Illex5*. This included the calculation of confidence limits, which we will use later under this section. Therefore you should make sure that *Illex5* is still the loaded fit (Use **Fit Manager** to Load the fit if necessary).

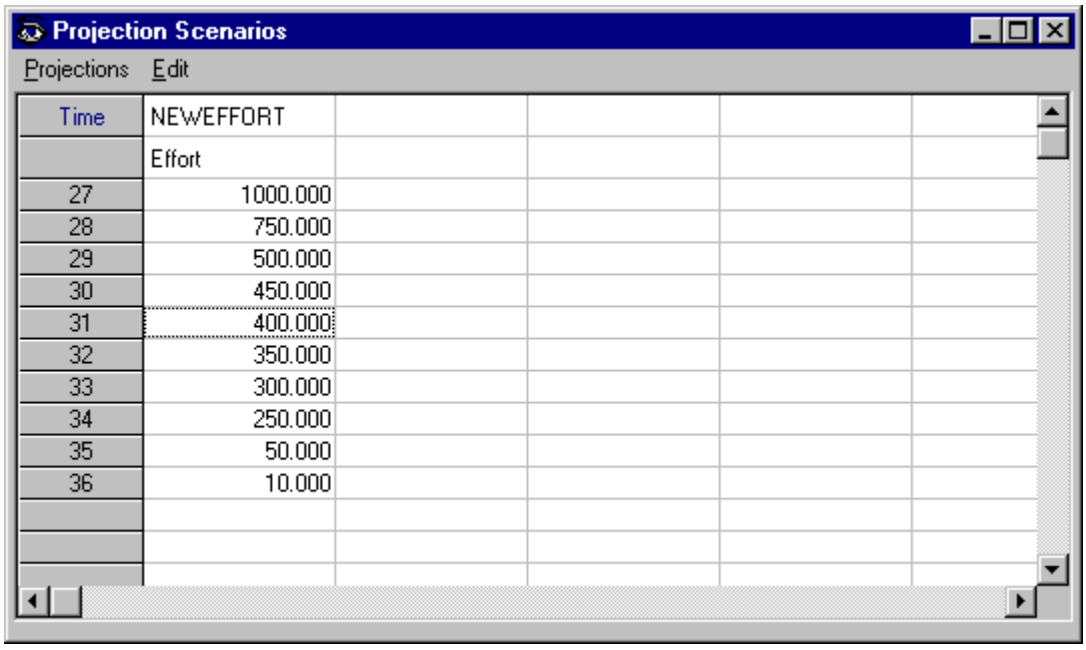
Select **Projections** from the menu bar. The only option available on the pull down menu at this stage is **Set Up Scenarios**. The other options become available once you have added your first projection scenario. Select **Set Up Scenarios**. A new window appears with the title "Projection Scenarios" and an eye for an icon. The Projection Scenarios window contains a data sheet which is used to input and store the projection scenarios. The leftmost column will contain the timing for the future projections. To the right of the timing column, each column can hold an individual Projection Scenario. The top row in each scenario contains the unique descriptor (i.e. the title) for that scenario. By default the first scenario is given the title "Scenario1". The default title can be edited by double clicking on it with the and then typing the new title. We suggest you change the title to NEWEFFORT and press <return> In the next row down you must specify the type of projection scenario you are going to create. The example we use here is a projection of fishing effort. Effort is given by CEDA as a default, however, if you double click on Effort you will see a pull down list of options appears. You should select Effort and press return.

The cursor should now be positioned under the heading NEWEFFORT (effort). You are going to enter some numbers in this column which represent possible future levels of effort in the fishery. The column to the left is the timing column. This will be filled automatically as you enter the effort figures. The first row of the timing column is already completed (week 27) because this follows on from the squid dataset on which the projection is based (the last row in the dataset was week 26).

We will be projecting the fishery for a further 10 weeks, up to week 36. The projected effort will start at 1000 effort units and decline to 10 effort units over this period. CEDA assumes that the effort units in the projection scenario are the same as in the original dataset.

There are two commands, or 'hotkeys' available for entering the projection scenario. F4 adds a row and F5 deletes a row. At the start the first row of effort data contains 0.000. Press F4 and a second row will appear with 28 in the timing column (representing week 28) and 0.000 in the NEWEFFORT column. Press F4 eight more times, until the number 36 appears in the timing column. You can also double click below the most recently added time interval to add an additional row. You can now use the up and down arrow keys or the mouse to move up and down the [NEWEFFORT] column. Move to the top row. Type 1000 and then press the down arrow key. The number 1000 will appear in the top row and the cursor will move to the second row. Now enter the number 750 in the second row (week 28) and move to the third row by typing 750 and then the down arrow key. Complete the column of data by entering the following additional numbers: 500, 450, 400, 350, 300, 250, 50 and 10. The Projections Scenario WIndow should now look like that shown below in Figure 5.2.13.

Figure 5.2.13 The Projections Scenario Window



Close down or minimise the Projection Scenarios window and return to the main CEDA application window. Click on the **Projections** menu option again and you will see that the Project option in the pull down menu is now available. Select **Project** and a MultiPick dialog box will appear on the screen. The list of available projection scenarios is on the left hand side. So far you have only created one scenario - NEWEFFORT. You should select this from the list in the usual way (click on [>] and then on OK).

A graph appears on the screen with a blue line showing the population size over the period of the fishery, and a red line showing projected population size for the period week 27 to the end of week 36. If you had multiple projection scenarios selected then a number of lines would appear. If you want to see the projected trajectory more clearly you can opt to have the projection plotted by itself; do this by depressing the 'tick' button on the toolbar to turn off the fit line.

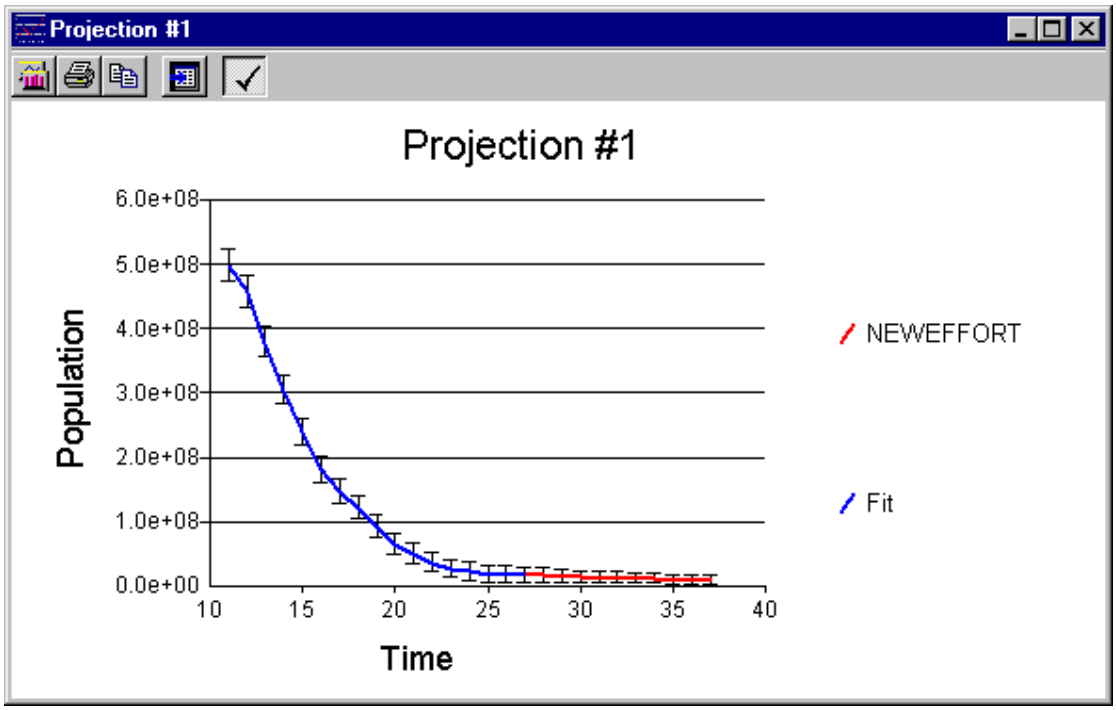
The graph has been given the default title "Projection #1". If you wish to change the name of the graph you can open up the Graph Control Editor by clicking on the left most button on the toolbar. From within the Graph Control Editor select the tab labelled Titles and change the title to one of your choice and select OK.

Note: All plots are accessed from the **Graph** menu by their title. If you change the graph title then its name will also change under the **Graph** menu.

Close down the projection graph and select **Projections | Project with Conf. Ints** from the menu. This enables you to do the same projection, but including the confidence intervals you calculated earlier (saved with fit *Illex5*). Note that you will only be able to use the option **Project with Conf. Ints** if the fit which is currently loaded includes calculated confidence intervals. Pick a scenario from the dialog box which appears. Again, NEWEFFORT is the only scenario you have created so far, hence there is only one available. This time you can only choose one scenario, rather than several, because the resulting graph includes error bars and plotting more than one projection at a time would lead to a very confused picture. Click on OK again. CEDA then prompts you to specify the confidence interval you want to use. The default provided is 95%. Accept this by clicking on OK. CEDA then spends some time generating confidence intervals. When it is finished, a graph similar to the previous one,

but with error bars representing the 95% confidence interval, appears on the screen (Figure 5.2.14).

Figure 5.2.14 Projection with Confidence Intervals



Adding a Scenario

If you return to the Projection Scenarios window again (by maximising it or selecting **Projections | Set Up Scenarios**), you can set up another projection scenario using different levels of fishing effort or catch. Select **Add Scenario** from the pull down menu (or double click anywhere to the right of the existing scenario). You must specify a new name for the projection scenario (each one must have a different name) and indicate the units of the projection (Effort, Total Catch Weight, or Total Catch Numbers]. You can enter the data using the keyboard as before. When you have finished, return to the main data window, and repeat the procedures described above to run the projection. Remember that this time you can select multiple scenarios from the MultiPick dialog box under the option **Projections | Project**, to plot more than one scenario at a time.

All of the projection scenarios you create are automatically saved in your CEDA file SQUID.CD3. They can be re-run any time you start CEDA and use the **File | Open** command to open the file SQUID.CD3

You have now completed your analysis of the squid data. In the next section we will go on to analyse the tuna data you imported earlier from the file TUNA.CD2.

5.2.2.6 Conclusions from the analysis of the squid data set

The no recruitment model with gamma error gives both satisfactory and useful fits to these data. The diagnostic plots (residual/percentile plots) do not highlight any major problems and the confidence intervals are narrow. The importance of considering the model assumptions carefully and using background data is illustrated by the arguments used to justify the exclusion of the first three data points. The remaining potential outlier suggests that there may still be some problems with the model or the data, but the lack of sensitivity of the parameter estimates to this data point indicates the utility of the model.

One important conclusion is that the parameter estimates are very sensitive to the value used for the natural mortality rate ***M***, and that the data yield very little information about ***M***. This is a common problem which arises when there is a lack of contrast in the data, making it very difficult to separate the natural and fishing-related components of total mortality ***Z***. Given this sensitivity, a sensible course for management might be to consider how to improve the estimate of ***M***.

5.2.3 Tuna tutorial

We will be working on the data set XTuna.cd3 which is located in the default CEDA directory (or the directory in which CEDA has been installed).

Before proceeding with the Tuna data analysis open this file using the **File | Open** menu option.

5.2.3.1 Selection of model, fitting and analysis

There are three columns displayed in the data window: Tot. Catch Wt, Effort and Catch Wt. The data in column Catch Wt are the same as those in column Tot Catch Wt. This is because the data in column Effort represent the total fishing effort for the whole fishery rather than for just part of the fishery.

Select **Fit | New Fit** from the menu. You will see that there are only three models available due to the types of data which you have loaded (catch in weight, and effort). They are all **Production Models: Fox, Schaefer and Pella-Tomlinson (PT)**. Select PROD. MODEL (PELLA TOM) from the list and click on OK. A dialog box appears on the screen requesting certain input parameters:

Initial Proportion

The degree of exploitation of the stock before the start of the dataset (see Deterministic Recruitment/Production (DRP) Models). Responses should lie between 0 and 1, with 1 indicating negligible prior exploitation, and 0 indicating a stock almost completely exterminated before the current dataset began. Prior exploitation of this tuna stock is thought to have been negligible, so enter **1** here.

Z Shape Parameter

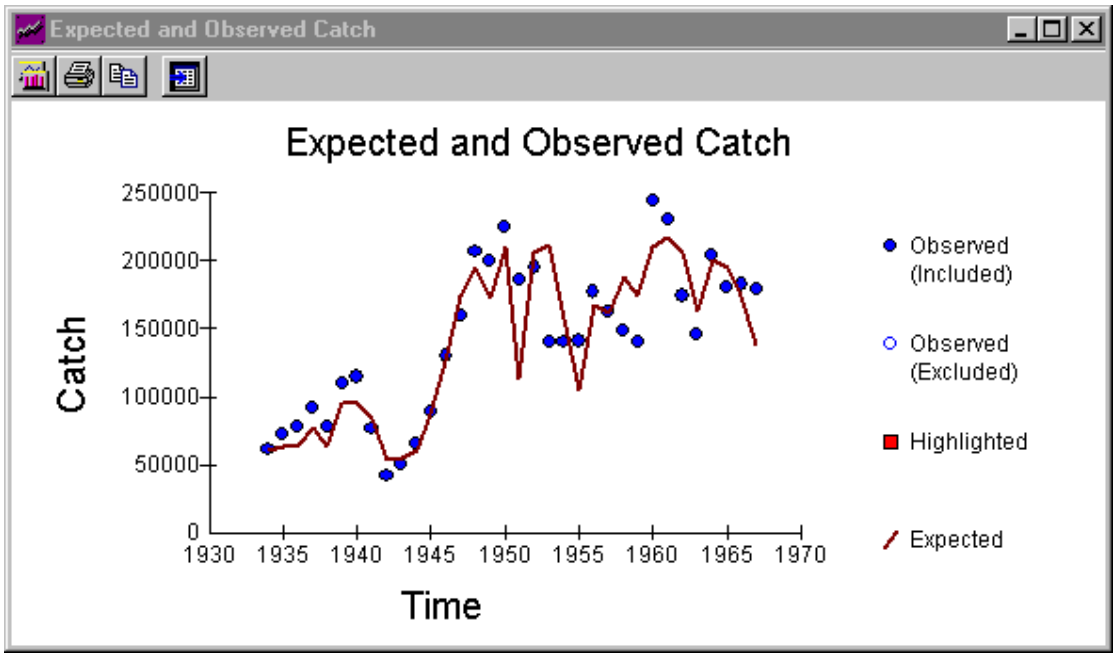
A skewness parameter applying only to the Pella-Tomlinson model (see Pella-Tomlinson Production Model); accept the default of **1.**

Time Lag

This is explained in Time Lags in Production Models; here you should enter **0**, which is usual for preliminary examination of a dataset

Select Least Squares (Unweighted) from the list of Error Models and leave the other Options as they are (neither selected). Click on OK and CEDA will fit the model. A plot of Expected and Observed Catch appears on the screen (see Figure 5.2.14), as it did when you analysed the squid data. The fit looks reasonable; most of the features of the data have been captured, but there are some periods when the observed data are all on one side or other of the fitted line, particularly towards the left-hand side of the plot.

Figure 5.2.14 Least squares fit to tuna data



As with the squid data, the next step is to identify the best error model. Use the **Graph** menu to display plots of Residual Catches vs Time and Residual Catches vs Expected Catches. There are patterns in both residual plots; the degree of scatter seems to increase with both time and expected catch, and there are two outliers (1951 and 1953). These patterns suggest that least squares may not be the best error model. Close down all graphs and return to the main CEDA application window. Save the fit in the usual way using Fit Manager, calling it *Tuna1*.

You should now examine the performance of the gamma and log transform error models. Select **Fit | New Input Parameters** from the menu. Use the same values for Initial Proportion, Z Shape Parameter and Time Lag as in the last fit (1,1 and 0 respectively). Repeat the analysis using the log transform model. Use Fit Manager to save your work after fitting. Call this new fit *Tuna2*. Remember that this fit are saved with the data file (XTUNA.CD3).

Now repeat using the gamma error model. You should now find that an unexpected error message comes up, telling you that the minimisation failed, with the further information that it "succeeded on function" but failed on "parameter". This is almost certainly completely incomprehensible, and in such circumstances you should try the Help button. This will direct you to the section dealing with the Options menu, where parameters for the minimisation are set (see Options menu). As explained there, the minimisation has converged properly in terms of successive changes in the function value being very small , but after 500 iterations the individual parameter values are still changing by too much, so that minimisation on the parameter values is deemed to have failed. If you reduce the Parameter tolerance value to 0.001, you should find that the convergence now occurs on both function and parameters. That is sufficient for now, but be sure you take heed of the advice regarding minimisation parameters when you are analysing your own data sets! Again, use Fit Manager to save your work, calling the fit *Tuna3*.

Have a look at the residual plots (percentile plots for the gamma fit) by selecting them from the Graph menu. There seems to be less pattern in the residual (percentile) plots for the log transform and gamma error models than there was in the least squares plot. However the outliers at 1951 and 1953 are still there.

There are now two questions to be answered:

- what should be done about the outliers?
- which of the two best error models should be used?

The dotted horizontal lines used as "outlier guides" in the residual (percentile) plots are set so that on average 1 data point in 20 will fall outside just by chance. There are 34 data points in this dataset. The fact that 2 lie outside the outlier guides is therefore not surprising. However, the two data points in question are actually quite a long way outside the outlier guides. This is obvious for the log transform model, but less so for the gamma because percentile plots are used. If you display just one percentile plot on the screen at a time, you should be able to see that the two outliers are very close to 0 and 1. They are certainly closer to 0.005 and 0.995, where outlier guides for 1 outlier in 100 good data points would appear, than they are to the existing "1 in 20" outlier guides.

These two outliers therefore suggest that there is a problem with either the data or the model. Unfortunately, there is no extra information available to indicate which might be the source of the problem. There is no reason to believe that these points are suspect, as there was for the first three points of squid dataset. Therefore there is **no good reason** to exclude them from the analysis.

Remember: Never omit data points from the analysis unless you have a good reason to do so.

There are patterns in the residual (percentile) plots of both the gamma and the log transform error model fits. Neither of the error models seem to result in the Pella Tomlinson model fitting the data very well in the first few years. The Residual Catches vs. Expected Catches plot for the log transform fit looks particularly bad, showing residuals decreasing with expected catch above the middle range of Expected Catch. The corresponding percentile plot for the gamma fit does not show such a pattern as strong as this one. On the basis of the residual (percentile) plots the gamma error model appears to result in the best fit to the tuna data

Assessing the Sensitivity to Outliers

The next step is to assess the sensitivity of the parameter estimates to the outliers. This should be done by running a number of fits using both error models and using all combinations of including/excluding the two outliers. In order to save time in the tutorial we have done these extra fits for you. The following table presents estimates of ***K*** (carrying capacity) from these fits:

Estimates of ***K*** from various fits

Fit	**Gamma**	**Log Transform**
All data	1.387E+6	1.880E+6
Without 1951	1.074E+6	1.416E+6
Without 1953	2.446E+6	2.598E+6
Without 1951 and 1953	1.703E+6	1.851E+6

Both error models seem quite sensitive to the outliers, particularly the one in 1953. The gamma is marginally more sensitive than the log transform. The ratio between the highest and lowest estimates of ***K*** for the gamma model is 2.446/1.074=2.28, while for the log transform model it is 2.598/1.416=1.83. Note that strictly speaking this sensitivity should be assessed on the basis of confidence intervals rather than point estimates (see Confidence Intervals). If this were your dataset, more investigation of these two data points and the reasons for the sensitivity would be a high priority. An appropriate course of action would be

to run subsequent analyses twice, first including 1951 but excluding 1953, and then vice versa. These are the combinations that give the greatest difference in parameter estimates. However, this would take too much time in this tutorial section. Instead, we will proceed with one set of analyses, including all the data in every fit.

The log transform fit seems slightly less sensitive to the outliers, however, there are noticeably worse patterns in the log transform residual plots than in the percentile plots of the gamma fit. If this were your own dataset and you were undertaking a full analysis, you should investigate the source of the sensitivity to the outliers and residual (percentile) plot patterns in both the log transform and gamma fits. CEDA includes additional diagnostic tools which can assist in this investigation, such as the facility to view the minimisation graphically (see the Reference and Operating Guide). However, for the purpose of this tutorial we will pursue a single line of analysis using only the gamma error model.

The next stage of analysis is to examine the sensitivity of the fit and the estimated parameters to the input parameters (Initial Proportion, Z Shape Parameter and Time Lag).

Initial Proportion

Investigate sensitivity to Initial Proportion by successively reducing the input value from 1.0 (the default) to 0.7. The stock is thought to have been less heavily exploited before the start of the dataset than after it. It is reasonable to expect the stock size (and hence expected CPUE) to fall rather than rise during the first few years of the dataset, in response to the increase in the level of exploitation. Using this criterion, and the evidence from the diagnostic graphs, you will find that values of about 0.9 and above lead to reasonable fits, and the parameter estimates are insensitive to changes over the range 0.9 to 1.0. We will therefore continue the analysis with Initial Proportion fixed at 1.0.

Time Lag

Next we suggest you investigate the effect of varying the Time Lag. According to the biology of the stock, time lags in the range of 0 to 4 could be reasonable. The following table shows the parameter estimates from fits with time lags of between 0 and 4, with all other input parameters remaining as they were for the original fit. You should try at least some of these fits yourself and examine the diagnostic plots to see which fits look best.

The effect of Varying the Time Lag on the Parameter Estimates

Time Lag	**R^2**	***K***	***q***	***r***
0	0.825	1.387 e+6	7.850 e-6	0.4585
1	0.828	1.614 e+6	6.877 e-6	0.3875
2	0.833	1.807 e+6	6.219 e-6	0.3442
3	0.838	1.816 e+6	6.269 e-6	0.3488
4	0.838	2.006 e+6	5.730 e-6	0.3078

The parameter estimates seem to be relatively insensitive to changes in Time Lag within this range, although there is some sensitivity in ***K***. There is little apparent change in the quality of the fit, except at a time lag of 4 when the percentile plot Percentiles vs. Expected Catches starts to show patterns. An appropriate course of action might be to conduct further analyses using Time Lag values of between 0 and 3, to develop a picture of the potential uncertainty in the parameter estimates. However, to save time in this tutorial we will restrict the following analyses to a Time Lag of 0. We will see later that the variation in the estimated parameters resulting from changes in time lag shown in the table are insignificant compared to the inherent uncertainty in the data.

Viewing the Minimisation Graphically

Before moving onto the next sensitivity analysis, you should note the negative correlation between ***r*** and ***K*** in the above table. If you re-do the fit (**Fit | New Input Parameters**) and use the option **View Minimisation Graphically** you will see a contour plot of ***r*** vs. ***K*** for the parameters you have specified. When prompted, you should specify the following values to fix the limits of the contour plot:

K: 1e5 - 6e6

r: 0.1 - 1.0

No. of points to plot: 40

Bear in mind that CEDA will take some time to evaluate the contours for the gamma error model, particularly if you are working on a less powerful computer. The contour plot displayed clearly demonstrates the negative correlation between ***r*** and ***K*** and illustrates the shape of the fitting surface over which the non-linear minimisation is searching. The progress of the minimisation algorithm is superimposed on the contour plot.

If you want higher resolution in the contour plot you can increase the Number of points to plot up to a maximum of 100, but this correspondingly increases the time taken to evaluate the contours.

Z Shape Parameter

The final parameter to be investigated is the Z Shape Parameter or skewness parameter ***z***. We have prepared a table of parameter estimates for a range of ***z***, but again you should run at least some of these fits yourself to study effects on the quality of the fit.

The effect of Varying the Z Shape Parameter on the Parameter Estimates

z	$\mathbf{R^2}$	***K***	***q***	***r***
0.5	0.834	1.379 e+6	8.126 e-6	0.791
0.8	0.827	1.395 e+6	7.988 e-6	0.536
1.0	0.825	1.386 e+6	7.822 e-6	0.459
1.3	0.820	1.410 e+6	7.561 e-6	0.375
1.6	0.816	1.420 e+6	7.337 e-6	0.326
2.0	0.810	1.476 e+6	6.959 e-6	0.273

There is little difference in quality of fit as ***z*** varies between about 0.5 and 2.0, and the estimates of ***K*** and ***q*** are not very sensitive to changes in ***z*** within this range. The estimate of ***r*** seems to be more sensitive than ***K*** and ***q*** to changes in ***z***, particularly at the lower end of the range. There is in fact a strong negative, correlation between ***r*** and ***z***.

Once again, an investigation of the underlying reasons for this sensitivity in ***r*** is outside the scope of this tutorial. During an analysis of your own data you would certainly want to look at it in more detail. However, there are some observations which can be made at this stage. The apparent insensitivity of the estimates of ***K*** and ***q*** and the quality of the fit (which we have seen is not very good anyway) over a wide range of ***z***, indicates that there is not enough information in the tuna dataset to allow ***z*** to be estimated with any reasonable degree of precision. This is similar to the problems involved in estimating the natural mortality rate ***M*** from the squid dataset.

5.2.3.2 Generating confidence intervals for the tuna data

The generation of confidence intervals using the bootstrap method requires the fitting process to be repeated 500 times (this can be changed to a different number of bootstraps under the **Options** menu). Depending on the error model being used, and the power of your computer, this can take a considerable amount of time. The gamma error model is unfortunately the slowest! To save time in this tutorial the CEDA data file on which you are working (XTUNA.CD3) contains a pre-computed set of confidence intervals for the gamma error model.

Using Fit Manager reload the fit called *gamconf* from the list of available logged fits. This logged fit contains confidence intervals generated using the whole tuna data set, with the following specifications:

Z Shape Parameter = 1
Initial Proportion = 1
Time Lag = 0
Gamma error model
Num. Bootstraps = 1000

View the graphs on the **Graph** menu to get an idea of the shapes of the distributions, and note the numerical confidence intervals from the Parameter Estimates window.

The conclusions drawn earlier about the sensitivity of the model to various assumptions should be considered in terms of these confidence intervals. For example the 95% interval for ***K*** ranges from 7.81e+5 to 5.72e+6. The changes in ***K*** resulting from the changes in time lag and ***z*** shown in the above tables are well within this range. Such changes can be thought of as insignificant when compared to the uncertainty inherent in the data. Similarly the 95% interval for ***r*** ranges from 0.040 to 0.875, which is greater than the variation in r resulting from changes in ***z*** of between 0.5 and 2.0.

The confidence intervals for the gamma fit are very large. The estimated Maximum Sustainable Yield (MSY) is about 160,000 tonnes, but the lower limit of the 95% confidence interval of MSY is only 55,000 tonnes. There is corresponding uncertainty in the estimate of replacement yield. Nevertheless, as with the squid data, these limits may not reflect the true extent of the uncertainty in the tuna data. A reasonable course of action for further analysis would be to generate a range of confidence intervals using different combinations of influential input parameter estimates and including/excluding influential points (e.g. the outliers in 1951 and 1953). A final conservative confidence interval can then be estimated by taking the highest upper limit and lowest lower limit over all of the resulting confidence intervals.

5.2.3.3 Conclusions from the analysis of the tuna data

No matter what parameter adjustments are made to the Pella-Tomlinson model, substantial discrepancies remain between this model and the tuna data. The residual or percentile plots show trends and runs of positive and negative residuals. There are also two unresolved outliers. It is clear that there are problems, either with the suitability of the model or with the data, which require further investigation.

In terms of the information content of the data, the confidence intervals produced in this analysis are wide, and the results are sensitive to the inclusion or exclusion of the outliers, as well as to some input parameters. There is no chance of estimating ***z*** with any precision from these data alone, but the value chosen substantially affects the estimates of ***r*** (although the range of ***r*** for ***z***'s of 0.5 to 2.0 is within the limits of the 95% confidence interval. The diagnostics in CEDA prove useful in eliminating certain possibilities, such as the least squares error model, low values of initial proportion and time lags of > 3 (background biological information rules out time lags higher that 4). However, it is difficult to draw firm conclusions,

because the information content of the data appears rather low, despite the number of data points.

This dataset is in fact an example of a "one-way trip", as discussed in Chapter 9 of Hilborn and Walters (1992). The lack of contrast in the data means that good parameter estimates would always be difficult to obtain. For a fishery manager interested in guidelines for management, such as MSY, it would be sensible to think about a number of ways of improving the analysis. For instance, there may be other ways of deciding what value(s) of ***z*** to use. Ultimately it might be more sensible to use a simpler production model such as Schaefer or Fox which have fewer input parameters. Methods of improving the contrast in future data from the fishery might also be considered. Some ways of doing this are discussed in part IV of Hilborn and Walters (1992).

6. Contribution of outputs

The revised CEDA and LFDA software will provide scientists and fishery officers in developing countries with access to modern Windows versions of the popular but dated Dos versions of these programs. Able to run in Windows 95 or later operating systems, both packages now have easy-to-use interfaces to other spreadsheet, database and graphics software that are commonly used. The revised packages will therefore enhance the use of the stock assessment methods embodied within them and thereby further facilitate development of sound scientific advice to fishery managers. Such improved scientific advice will enhance the likelihood of sustainable management of vital fishery resources, which in developing countries often represent major sources of animal protein, employment and income.

7. References

Hilborn, R. and Walters, C. J. (1992) Quantitative Fisheries Stock Assessment. Chapman & Hall.

Pauly, D. (1987) A review of the ELEFAN system for analysis of length-frequency data in fish and aquatic vertebrates. Pages 7-34 in Pauly, D. and Morgan, G. R. (Eds) Length-based methods in fisheries research. ICLARM, Manila, Philippines and KISR, Safat, Kuwait. 468p.

Rosenberg, A. A., Kirkwood, G. P., Crombie, J. A. and Beddington, J. R. (1990) The assessment of stocks of annual squid species. Fisheries Research 8:335-350.