

## **Additions to the CREWES Matlab Library**

Christopher B. Harrison

### **ABSTRACT**

This report will show and discuss the various Matlab programs that have been significantly altered and or created by CREWES Research Associate, Christopher Harrison. The development of these additions and modifications has mainly taken place in the PC environment in release 13 of Matlab. It has been attempted to limit, if not completely avoid the use of PC specific commands in each of the programs, but cross platform issues may still occur. This limiting of PC commands was attempted due to the lack of testing on UNIX based systems and no testing on Mac based systems. Testing of the programs covered in this paper has taken place, but field usage may reveal more issues than lab testing could ever find.

There are two programs that have not been changed, but simply rewritten to take advantage of new developments in the Matlab programming environment. The two programs in question are `askthings` and `namecolor` whose original routines are still functional and available in the CREWES Matlab library. Two new programs that have been added to replace these antiquated routines are called `askthingsle` and `namecolorle`. These new programs differ from their progenitors mainly in their usage of `uiwait` and `uiresume` commands which allow for calling their routines only once from other programs. `Askthingsle` is similar to its forbearer in that it is a dialogue that allows a developer to input questions with selectable or editable answers, and have a user return specific answers. `Namecolorle` is similar to its forbearer in that it is a dialogue that allows a developer to question a user to change the name and or color of a specific variable. It is encouraged that `askthingsle` and `namecolorle` be used when developing new routines to move away from their antiquated counterparts. The simple routine, `figcent`, has also been changed to allow new figures to be centered not just on the main screen, but also on specific figures.

A geophysical program added is called `lascreator` and allows user to create simple synthetic Version 2.0 LAS files. The GUI displays four blocks that allow a user to input four characteristics, Depth, Vs, Vp, and density into a maximum of 99 layers. Depth can be chosen as feet, meters or seconds, density has 6 separate units, with Vs and Vp being entered as standard velocities depending on the depth units. A plot is displayed on the `lascreator` window to show relative values of characteristics and present location of the 4 visible layers. This routine can be used through the command window of Matlab or in conjunction with other programs.

A non geophysical program has been added called `matinterrogator`. This routine offers an easy way that a user can select, rename, and change information in arrays of a selected `.mat` file. The program can be used from the command window or integrated into a program. `Matinterrogator` can be used on its own from the Matlab command window or in conjunction with other programs.

## ASKTHINGSLE

Askthingsle is a quick dialogue that allows developers to ask users question with specific selectable or blank editable answers that either are optional or required. The original routine askthings was developed by Dr. Gary Margrave and still available in the CREWES Matlab library. While the older version askthings is still available, it is encouraged that any developments now include the new version askthingsle and not the older original. Askthingsle is similar to its predecessor but utilizes newer Matlab functions that allow askthingsle to be called once rather than 2 separate instances which was the case with askthings. Askthingsle can be opened with various arguments in the following ways.

```
ansfini = askthingsle ( masterfig , qst , a , flags );  
ansfini = askthingsle ( masterfig , qst , a , flags , titlestr );  
ansfini = askthingsle ( masterfig , qst , a , flags , titlestr , ttstr );  
ansfini = askthingsle ( masterfig , qst , a , flags , titlestr , ttstr , figmode );
```

The first way of initializing askthingsle shows the least amount of arguments that are needed to open askthingsle properly. The 4 arguments *masterfig*, *qst*, *a*, and *flags* are described below

***masterfig*** – This argument is the handle to the figure that has called askthingle. Since askthingsle is meant to be used in conjunction with other routines a master figure is needed to “reassure” askthingsle that its code is being used properly.

***qst*** – Is a cell array of the questions that a user is asked. The question cell array must be set up with one row and any number of columns with each column containing a single question that is desired to be asked.

***a*** – Is a cell array of the answers for the cell array questions. The answer cell array must be set up with one row and the number of answer columns being equal to the number of questions. There are two answer formats available, an editable answer which is either blank or predefined, or a popup menu with multiple predefined un-editable choices. An editable answer is simply a predefined string, empty or otherwise, that will appear as being a required answer or an optional answer depending on the *flags*. A multiple predefined popup menu is created by placing a ‘|’ between answers in a single string array of the answer cell (“Up|Down|Left|Right”).

***flags*** – This numerical array controls whether the editable string answers that have been passed to askthingsle are optional or not optional. The values of the flags are either 1 or 0, corresponding to non-optional and optional answer respectively. The number of flags has to equal the number of answers with the flag array being set up as a numerical array with each column corresponding to a respective answer. The flag values do not affect the popup answers (yet) but flag values are still needed for the proper operation of askthingsle.

The arguments *titlstr*, *ttstr*, and *figmode* are not necessary to the operation of *askthingsle*, but do offer a slightly more in-depth control of the *askthingsle* figure. Each of these arguments is described below.

***titlstr*** – This is a title string which will appear near the top of the *askthingsle* figure window. The argument is past as a single string for simple help instructions or any other purpose the developer wishes.

***ttstr*** – Is a cell array of tool tip strings for the questions that have been asked. A tool tip string is the automatic help string that pops up on a user’s cursor when a question string on the *askthingsle* figure window has been moused over. The strings are meant to display simple help instructions or other information for each question. The tool tip string cell array must be set up with one row with the number of string columns equaling the number of questions that have been asked.

***figmode*** – This argument controls whether the *askthingsle* figure is modal or not. It is defaulted to 1, or none-modal, but can be changed to 0, modal.

Running *askthingsle* with 7 question and answer pairs and a title string “Plot Image Properties” results in figure 1.

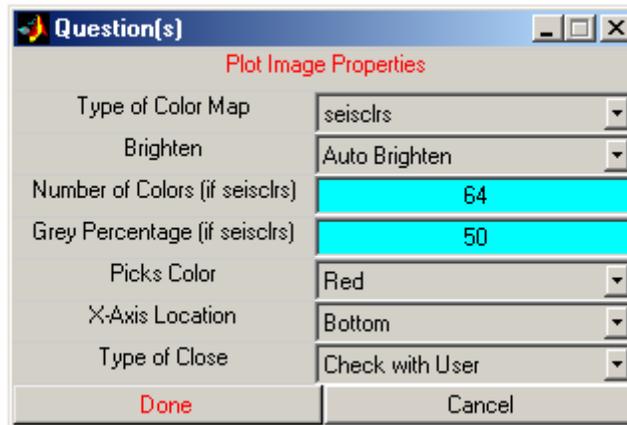


FIG. 1. The figure shows *askthingsle* figure window with 7 questions.

*Askthingsle* can be completed by pressing “Done”, “Cancel” or simply closing the figure window. When any of these options has occurred the final argument *ansfini* is passed.

***ansfini*** – Is a cell array of answers that the user has chosen in the *askthingsle* figure window. Each cell corresponds to an answer that was originally passed when *askthingsle* was first executed. If the *askthingsle* figure is closed, an empty cell array is passed.

### NAMECOLORLE

This routine is used when dealing with renaming and the color change of line objects. Again, similarly to *askthings*, *namecolor* was originally developed by Dr. Gary

Margrave. The routine was rewritten and renamed to `namecolorle` to include new Matlab developments that increase the speed and usability of the program. `Namecolorle` needs to be called only once from other routines rather than two times with the older `namecolor`. While the older `namecolor` routine is still available in the Matlab library, it is encouraged that `namecolorle` is used in new developments. The following shows the various ways to execute `namecolorle`.

```
Ansfini = namecolorle ( masterfig , nmqst , nmans , colorqst , colorans );  
ansfini= namecolorle ( masterfig , nmqst , nmans , colorqst , colorans , titlestr );  
ansfini = namecolorle (masterfig, nmqst, nmans, colorqst, colorans, titlestr, optstr, optlist);
```

The first way to call `namecolorle` requires only the necessary arguments for the proper execution of the routine. These arguments *masterfig*, *nmqst*, *nmans*, *colorqst* and *colorans* are described below.

***masterfig*** – This argument is the handle of the figure that has called `namecolorle`. `Namecolorle` is meant to be used in conjunction with other routines and `master figure` is needed to “reassure” `namecolorle` that its code is being used properly.

***nmqst*** – A simple string question pertaining to the name which is desired to be changed.

***nans*** – This argument is the name of the item to be renamed or simply displayed. Simply displaying the name is a new feature that has been added to `namecolourle` to limit the routine to changing color only. To display name only without the possibility of editing, place a “\_” or an underscore at the beginning of the name in question. This will stop `namecolorle` from displaying an edit box for a new name to be entered.

***colourqst*** - A simple string question pertaining to the color change.

***colorans*** – A Matlab color matrix of the initial color for the user to change.

***titlestr*** – This is the title that will appear at the top of the `namecolorle` figure. This argument is not a minimum requirement for `namecolorle` but it can be used to display help instructions for the user.

The last two arguments that are allowed to be passed in `namecolorle` are a remnant from the original `namecolor`. When these two arguments *optstr* and *optlist* are input, a new set of buttons is added to the `namecolorle` figure.

***optstr*** – This argument is a single string question that will appear on the `namecolorle` figure as a check box option. When the user checks the check box which corresponds to the question, the `namecolorle` with figure will change and offer the option list popup menu.

***optlist*** – This argument is the string answer popup menu that is displayed when the user has checked the optional question check box. The argument is a string with “|” separating the various answers that the developer wishes the user to choose from.

Figure 2 shows the namecolorle figure with an editable name on the left, the selected option check box and new layout for the option questions on the right.

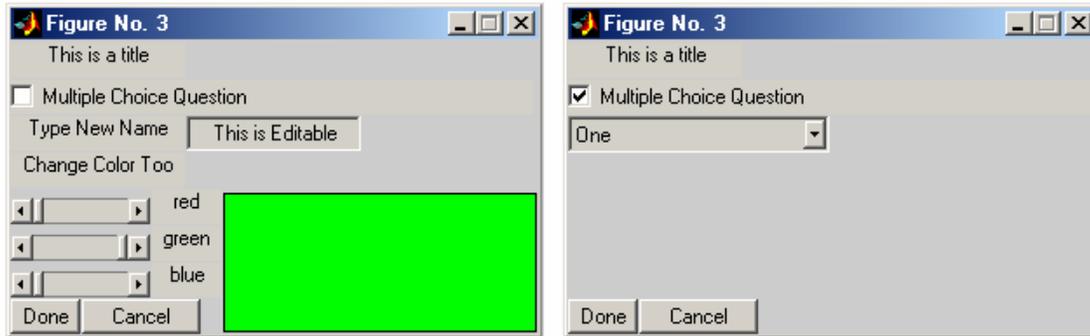


FIG. 2. The namecolorle figure on the left shows the options to edit a name and edit a color. When the check box is checked, the namecolorle figure will change to look like the figure on the right.

The initial color appears in the axes as seen on figure 3 and adjusting the slide bars marked “red”, “green”, and “blue” will change the color accordingly. When the name is set to be un-editable no edit box will appear, only the string that has been entered after the underscore.

*ansfini* – Is the array that is released when namecolorle is continued or closed. When continuing, namecolorle passes answers as three cells, the first being the answer to the name question, the second being the answer to the color question and the third being the answer to the optional question. If namecolor is closed or canceled, a blank cell is returned.

Future developments in namecolorle will integrated the optional questions better into the GUI.

## MATINTERROGATOR

This program allows a quick way for users to open .mat files and select for export or saving only the arrays of interest in those files. This program can either be accessed through the command window or be used as an uninterruptible file interrogation tool in conjunction with other program. The following shows the three ways matinterrogator can be accessed.

```
dataout =matinterrogator
dataout = matinterrogator ( file );
dataout = matinterrogator ( file , n );
```

When matinterrogator is executed with no arguments the standard file selection window is opened where the user can simply select the desired .mat file. An error message will appear if user tries to open anything other then a .mat file. The following explains the arguments that can be passed directly into matinterrogator.

*file* – This argument is the file location and name of the desired .mat file.

*n* – This argument is the number of arrays that will be allowed to be selected and exported by the user. Default for this argument is 40 arrays, which also corresponds to the largest number of arrays that matinterrogator can display.

Figure 3 shows what the matinterrogator window looks like when opening a .mat file with three arrays.

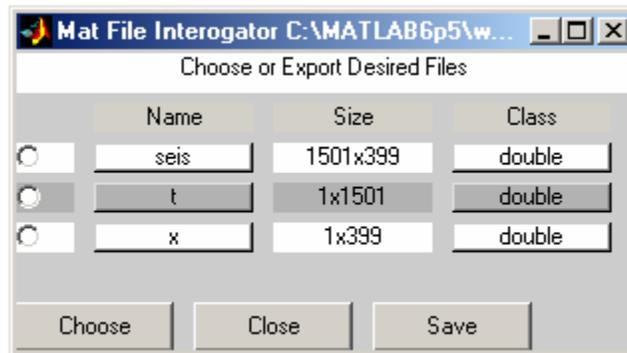


FIG. 3. The matinterrogator figure window showing three arrays from a .mat file.

The names of the arrays and class of arrays appear as push button functions in the matinterrogator window. The name of the array can be changed by pressing the name push button of the desired array. This will pop up a dialogue asking the user to rename the array as desired. Pressing the class push button of an array will have different effects depending on what type the array is. If the array is a string array, matinterrogator will allow for the user to edit the contents of the array. If the array is numerical and shorter than 4 rows or 4 columns, again matinterrogator will allow the user to edit the array manually. If the numerical array is bigger than 4 rows or 4 columns, plotting options will be asked so users can inspect, to a certain degree, the array in question. It should be noted that special characters can not be used in the new naming of arrays.

If the class of the array is a cell or structure the arrays are treated similarly to .mat files and opened in a new matinterrogator window. Cells names are listed by number and the structures are listed by their field names. This functionality will continue till there are no cells or arrays to be opened in the respective file.

Arrays are selected by using the check box on the far left of each listed array. When an array is selected, the entire row of info will be highlighted orange to ensure the user knows which arrays are selected and which arrays are not. If the argument *n* has been passed, only *n* arrays will be allowed to be selected for exporting.

Depending on how the matinterrogator has been called, the user will be asked to either "Choose" the files, when called from the command window, or "Export" when used in conjunction with other programs. The option to "Save" will also only appear when matinterrogator is called not using an argument.

***dataout*** – This argument is passed whether data is exported or the matinterrogator figure is closed. When the window is closed *dataout* is passed as an empty array. Otherwise, the *dataout* is a structure that contains the arrays in order that they were selected.

Using save allows users to save only certain arrays in a .mat of their choice to a new .mat file. When saving is complete, the matinterrogator window is closed without passing anymore arguments.

Future changes in matinterrogator will allow for essentially an unlimited number of arrays to be displayed rather than the 40 which it is limited to now. More control over plotting options, and more in depth ability to manipulate numerical arrays is also under consideration.

### FIGCENT

This is a very simple program that has undergone a major renovation without destroying its functionality for older applications. The program enables a simple way to center new specifically sized figures on a main monitor with the new added ability to allow for centering of these new figures on master figures. The program also takes into consideration the size of the new figure in relation to the size of the screen and will not allow the new figure to be larger than screen area or appear off the screen.

```

newfig = figcent
newfig = figcent ( width , height )
newfig = figcent ( width , height , units )
newfig = figcent ( width , height , units , masterfig )

```

When *figcent* is called by itself, a new figure is created half the width and half the height of the main monitor. The following explains the arguments that can be past through *figcent*.

***width / height*** – These arguments must be entered together and represent the width and height of the desired new figure. The default units for the height and width are normalized.

***units*** – At this point there are only two units that can be chosen for this argument, pixels and normalized. The *width* and *height* should be chosen in respect to the *units* passed.

***masterfig*** – This optional argument must be a handle to a desired master figure. This argument allows for the new figure to be centered on the master figure rather than the main monitor.

### LASCREATOR

This program allows users to create simple synthetic LAS 2.0 files on their own through a GUI. The GUI allows the user to add Vs, Vp, density and depth as well various other information to their own LAS well logs. The program is designed to work straight off the command window as well as in conjunction with other programs. The following shows the two ways that *lascreator* can be called.

Dataout = **lascreator**  
 Dataou = **lascreator** ( 'Open' )

Either way of calling lascreator can work in the command window, but the second way with an argument initiates the uiwait and uiresume section. This way lascreator will be uninterruptible when called from another program. Figure 4 shows the layout of lascreator with 4 layers added.

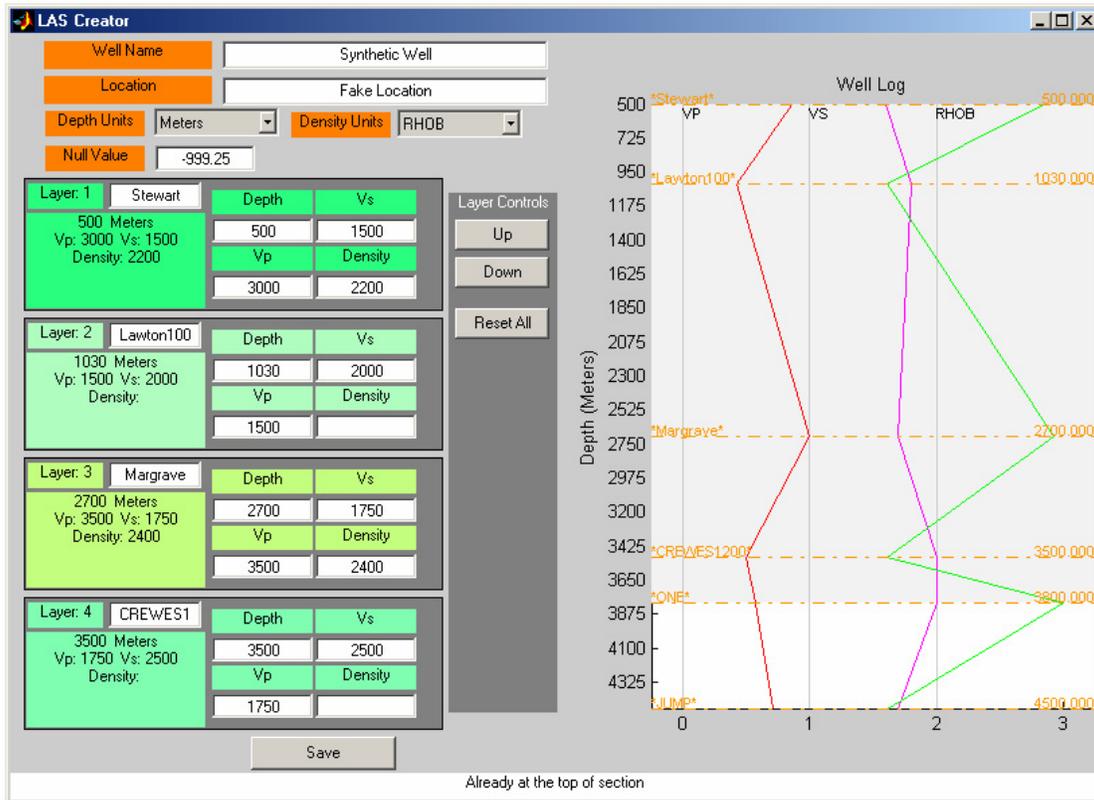


FIG. 4. The GUI for the lascreator program.

A maximum of 99 layers can be created with lascreator. The information in each layer is typed into 4 blocks that are located on the left side of the lascreator figure. The values for Depth, Vs, Vp and Density are regularly checked by lascreator and none numerical input cleared. Each block has separate controls that are accessed through a right mouse button click. Figure 5 shows the list of expanded options that appears for each block to control data for each layer.



FIG. 5. When a right click on any of the 4 block areas, the above set of options appears.

The two expanded options “Shift Data:” and “Insert Layer:” of course can not appear simultaneously as figure 5 shows. Shifting Data either up or down will move all layer information, excluding depth, to the layer above or below respectively. Clearing Layer will simply remove all information from the block while still keeping editable boxes. Delete layer will removed the layer information and shift all the layers accordingly to fill in the new blank location. Insert Layer allows the user to place a new blank layer either above or below the respective layer.

The layers can be cycled through by using the “Layer Controls” of “Up” and “Down” located on the center of the lascreator figure. Scrolling will be stopped if the depth noted in block 1 and 2 are not greater then the depth noted in block 3. Lascreator will only display layer information that have inputted depths and above the deepest depth that has been entered. As cycling of the layers occurs, a darker gray rectangle appears on the axes to show the user where the displayed layers are located. All data, including hidden information, can be reset by pressing “Reset All”.

The main axes show the relative values for the information added in the layer blocks. The lines are not supposed to be detailed representations of the data, only reference points for users to see trends in their data. At present, the displayed layer information is Vp, Vs and Density. Depth can be selected to be meters, feet or seconds. At this point, density can be selected in 6 different forms, RHOB, RHGF, RHGA, PHID, DPHI, and DPSS. Each layer top is displayed with its name on the left and its depth displayed on the right of the depth line. If any Vp, Vs or Density line is clicked, the block information will change to highlight where the user has clicked. The darker grey rectangle will also shift on the axes to show the user where the displayed layers are located.

Other data such as “Well Name”, “Location” and “Null Value” can be changed at the top of the lascreator window. More in depth information can be accessed and edited by right clicking the figure area of the window revealing figure 6.

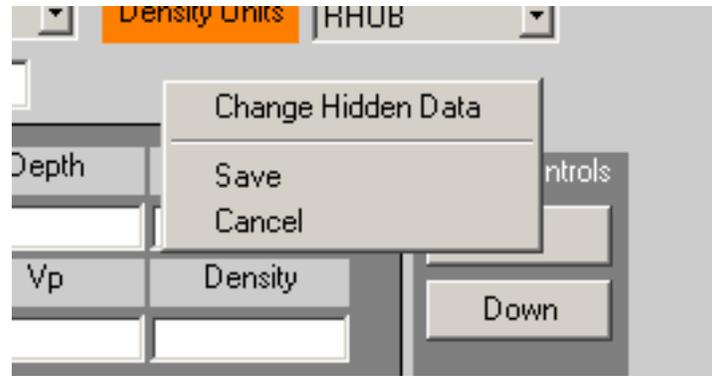


Fig. 6. Right clicking in the figure area of lascreator figure window gives the above options.

Clicking on “Change Hidden Data” allows the user to input new information for the *Company*, *Field*, *Service Company*, *Country*, *Depth Step*, *Height of Kelly Bushing*, and *Unique Well ID*. These values have defaulted answers and are accessed this way due to the fact that they are not necessary for the creation of a synthetic LAS file. The ability to save information and cancel is also available through this menu with each option acting similarly to its figure button counterpart.

Depending on how lascreator is called two figure buttons can appear at the bottom of the lascreator window. When lascreator is called without an argument, only one option “Save” is presented. Saving allows the user to save the information they have input as a version 2.0 LAS file. If lascreator is called with an argument, “Export” and “Cancel” buttons appear. Pressing either of these buttons will close lascreator and pass the final argument, *dataout*.

***dataout*** –If the user has chosen to “Export”, the *dataout* array is a cell with two values. The first cell contains the LAS Header and the second contains the log data. If “Cancel” button is pressed or the lascreator closed, an empty cell is returned.

When data is being exported, lascreator changes the Vs and Vp from the inputted velocities to the appropriate slowness. The density remains the same with the respective units chosen.

Lab testing has been done on lascreator and minor bugs worked out. Field testing may bring to light issues that have not observed before. Future developments under consideration are adding more input parameters, the ability of users to add their own parameters and the ability to open LAS files and make available to edit all parameters in the user chosen LAS file.