

## Starting *LINEAR*

You can start *LINEAR* from DOS by entering the command '`linear`'. In order to be able to use the '`Edit`' command, use '`lin`' instead.

While the program is running, the file `LINEAR.EXE` will not be needed again; however, for on-line help or error messages and explanations `LINEAR.HLP` will be needed. If you use the '`Edit`' command, your text editor must be present, and `LINEAR.EXE` must be present when you finish editing.

The program will look for `LINEAR.HLP` first on the drive and directory that was current when it was started, and then (if they have been changed) on the new current drive and directory. This could cause a problem if, e.g., the `LINEAR` files are in directory `\XXX` (which is on the command path) and you enter '`linear`' when directory `\YYY` (which might contain your data files) is current. There are three possible solutions:

- Keep a copy of `LINEAR.HLP` in directory `\YYY`,
- Start *LINEAR* with directory `\XXX` current and then use the '`Files Path`' command to change to `\YYY`, or
- Keep a configuration file `LINEAR.CFG` (see the next section) in directory `\YYY`, indicating `\XXX` as the path for `LINEAR.HLP`.

On a dual-floppy system, you might keep a disk with the *LINEAR* files in drive `A:`, and use drive `B:` if necessary for data files. On a single-floppy system, if you want to avoid switching disks you should keep a copy of `LINEAR.HLP` on your data disk.

If you wish, you may have *LINEAR* load a file immediately by including the name of the file on the command line (e.g. '`lin example1`'). The default extension '`.PRB`' is added if no extension is specified, so this example would load the file `EXAMPLE1.PRB`. `DIF` or `TAB` files can also be loaded immediately if you give a name with the extension '`.DIF`' or '`.TAB`'.

Input redirection can also be used on the command line, as outlined below. Output redirection on the command line will not work, as *LINEAR* does not use the DOS standard output. Instead, use the '`File Record`' command within *LINEAR*.

## Configuring *LINEAR*

When *LINEAR* begins, it looks for a file named `LINEAR.CFG` in the current directory. If present, this file determines several settings for *LINEAR*. If you are satisfied with the default values, you do not need a `LINEAR.CFG`.

The sample `LINEAR.CFG` shown below is included on the distribution disk. It uses the default values of all the settings. The file can be edited (with any text editor) to configure *LINEAR* for your needs. Each line of `LINEAR.CFG` should begin with the value to be set, followed by spaces and a comment. You may change the values, but do not insert spaces before them or change the arrangement of the lines.

---

```
C080  video mode
819   dialog buffer lines
.     path for LINEAR.HLP
.     path for data files
```

```
4      decimals
ON     show pivots
ON     keep artificials
ED     editor
```

---

The first line of the file starts with the display mode.

**C080** (80 columns  $\times$  25 rows color text) is the default. The only things actually displayed in color are inputs from the keyboard, which are yellow. On some black-and-white monitors attached to a color/graphics adapter, this color produces a mottled effect.

**BW80** (80  $\times$  25 black-and-white text) remedies that problem. This change has no effect if you have a monochrome or EGA card.

**EGA** or **VGA** selects 80  $\times$  43 mode on an EGA or 80  $\times$  50 on a VGA. Note that **C080** is still available on EGA and VGA, and will probably look much better, but if you want to pack as much information as possible on the screen you can use this option.

**LINEAR** will not work in 40-column modes.

The second line of the file indicates the size of the dialog buffer, in lines (see **Scrolling** below). This must be at least one screen (25 lines, except in EGA and VGA modes), and at most 819 lines. However, if not much memory is available, **LINEAR** may reduce the size of the buffer.

The third line of the file gives the path where the help file **LINEAR.HLP** may be found. “.” indicates the directory that is current when **LINEAR** begins. If, for example, the **LINEAR** files are kept in a directory **LINEAR** on drive **C:**, you might change this to **C:\LINEAR**.

The fourth line of the file gives the path for data files to be loaded and saved. This may be changed during the **LINEAR** session with the **'File Path'** command.

The next three lines of the file are for settings which are discussed in the section **Options** in Chapter 5.

The last line of the file gives your editor command. See the next section.

## Batch files and editing

In order to use the **Edit** command to temporarily leave **LINEAR**, edit a problem file, and then return to **LINEAR** to load that file, you must use batch files rather than running **LINEAR** directly from DOS. Included on the distribution disk are the two batch files **LIN.BAT** and **LIN1.BAT**. From DOS, enter the command **'lin'**. The current default disk drive must be one on which temporary files may be written (not a write-protected floppy, or a read-only network drive).

**LINEAR** needs the name of your editor in order for this facility to work. This may be specified in the configuration file **LINEAR.CFG** or in the 'Options' menu. For example, if your editor is **ED.EXE**, and you would use the DOS command **'ED F00'** to edit a file named **'F00'**, you should specify the name as **'ED'**. A drive and path may occur as part of the name. However, if your editor command is actually a batch file **ED.BAT**, you must specify the name as **'CALL ED'** (and this will only work if you are using DOS 3.3 or higher).






If you are familiar with DOS and batch files, you may wish to modify the batch files LIN.BAT and/or LIN1.BAT to suit your system. **The rest of this section is intended for those who are well acquainted with DOS and batch file programming. If you aren't, skip it or consult someone who is.**



The batch files work in the following rather circuitous way. LIN.BAT first calls LINEAR.EXE. If the 'Edit' command is used, LINEAR will produce a batch file &&TEMP.BAT (containing the commands needed to edit a problem file and reload LINEAR with that file), and exit. The DOS "errorlevel" is set to 210 to signal that this has been done (otherwise "errorlevel" is set to 0 on a normal exit from LINEAR). Unfortunately, LIN.BAT can't simply call &&TEMP.BAT: problems would arise if LINEAR produced another &&TEMP.BAT while the first is executing. Instead, when errorlevel 210 is detected LIN.BAT transfers control to LIN1.BAT, which first copies &&TEMP.BAT to &TEMP.BAT and then transfers control to the copy.







The first two lines of the batch file &TEMP.BAT call your editor with the name of the file to edit, and then call LINEAR to load that file. On the third line, after LINEAR exits, another test for errorlevel 210 is made, to determine whether 'Edit' was chosen again. &TEMP.BAT has six replaceable parameters, which are supplied by the program that calls &TEMP.BAT (in this case LIN1.BAT). The first three parameters form a DOS command that will be used if the test is true, and the last three parameters form a DOS command that will be used if the test is false. LIN1.BAT supplies only one parameter '%0' (which gets replaced by the name of LIN1.BAT itself). Thus if "errorlevel" is 210, the cycle repeats, and otherwise we return to DOS. However, if you wanted to use LINEAR as part of a menu system, you could change LIN1.BAT to supply a command line which would return control to the menu system. For example, in a system using AUTOMENU, the second line of LIN1.BAT might be



```
&temp %0 - - auto c:\math\math.mdf
```

## Using the keyboard

LINEAR is built around a system of menus, which appear at the bottom of the screen. Each choice will have one capitalized letter (usually the first), and one of the choices will be in reverse video (black on white instead of the normal white on black). There are two methods of making a choice in a menu: either press the key for the capital letter of your choice, or use the  and  c r s r- on ro ke s t mo e t e r v e s e v i e o a r o y u r h o c e n d r e s . I t h b a do s n t m v e y o p r b a l y a v  on: pressing the  key once should correct the situation.

In addition to the normal horizontal menus, there are vertical menus for choosing a file and in the 'Options' command. Here the  and  cursor keys move the bar. For choosing a file, the moving bar method must be used.

The key combination   acts as a toggle switch for echoing screen output to the printer (LPT1). The menus themselves or past screen output will not be sent to the printer. Pressing   once turns this feature on, and the next   turns it off. When printing is on, the letter P appears in the lower right of the screen. Printing can also be turned on and off with the 'Options' command.

The key combination   sends a "snapshot" of the current screen to the printer, as in most other programs.

To exit to the main menu from any other menu or when asked for input, press the **Esc** key.

On-line help is available. Press the function key **F1** to see a brief explanation of the menu item that is currently highlighted, or the input that is being requested.

## Line editing

When you are asked for input from the keyboard, the following keys will be useful:

- Home** moves the cursor to the beginning of the input area.
- ↑** recalls the last line that you entered. The last eight lines are remembered, and may be scrolled through with **↑** and **↓**.
- ←** (the cursor control key, not Backspace) moves the cursor to the left.
- moves the cursor to the right.
- End** moves the cursor to the end of the input area.
- Del** deletes the character the cursor is on.
- ←** (the Backspace key) deletes the character to the left of the cursor.
- Ins** toggles between “insert” and “overwrite” modes. In “insert” mode (which is the default), any characters at and to the right of the cursor are pushed to the right as you type. In “overwrite” mode, they stay in place and are replaced by what you type. The cursor is fatter than usual in “insert” mode, and thinner in “overwrite” mode.
- ←↓** enters your response. If the prompt contains something in square brackets, that is the “default response” that will be used if you just press **←↓** without typing anything.




## Scrolling

You will often find yourself wanting to see again something that has scrolled off the screen. LINEAR allows you to do this, by keeping the past and present screen contents (except for menus) in a dialog buffer. In any horizontal menu, the **↑** and **↓** keys scroll the screen back and forth one line, while **PgUp** and **PgDn** move it back and forth one whole screen. **Home** and **End** move to the beginning and end of the buffer. The dialog buffer holds up to 819 lines. Its size can be set in `LINEAR.CFG` (see the section **Configuring LINEAR** in this chapter). Before writing any new data to the screen, LINEAR automatically moves to the end of the buffer.

On a large problem, some of the commands can produce a large amount of output. Pressing **Ctrl S** or **Ctrl Num Lock** should stop the flow until another key is pressed (there seems to be some variation between machines here: **Ctrl S** does not always work). Pressing **Ctrl Break** should stop whatever is happening and return you to the main menu. You may wish to use the ‘File Record’ command to send the output to a file, where you can examine it at your leisure.

## Using a mouse

If your computer is equipped with a mouse, you may wish to use it rather than the keyboard to make choices in menus. The file `LINMOUSE.DEF` on the distribution disk is a menu source file which can be used with Microsoft or Logitech mouse systems. It must be compiled into `LINMOUSE.MNU` by Microsoft's `MAKEMENU` or Logitech's `NEWMENU`. When this menu is installed, moving the mouse will be equivalent to pressing the cursor control keys. The mouse buttons will work as follows:

<i>Microsoft</i>	<i>Logitech</i>	<i>Logitech Series 2</i>	<i>Action</i>
<i>two-button</i>	<i>three-button</i>	<i>two-button</i>	
left	left	left	
right	middle	both	
both	right	right	

You may modify `LINMOUSE.DEF` if you wish. See your mouse manual.

## Input redirection

Input redirection is the DOS facility that causes a program to take its input from a file or device rather than from the keyboard. It is invoked by placing '`<`' and the name of a file on the command line, e.g. '`linear example1 <solve.arf`'. I use the extension '`.ARF`' to stand for "Automatic Response File". This file must contain all the keystrokes that you would type at the keyboard in the `LINEAR` session if input redirection was not being used.


*WARNING: If something unexpected happens during the session, you have no way to regain control from the keyboard.*

This makes input redirection somewhat hazardous, but if you know exactly what you want `LINEAR` to do it can be a useful feature, as it allows automatic or "batch mode" operation. Here is my sample `SOLVE.ARF`:

---

```
FRex1
RSVV
```

---

To see how this does what it does, enter '`linear example1`' and type these keys yourself. After `EXAMPLE1.PRB` is loaded, '`FR`' makes the menu choices '`File`' and '`Record`', and '`ex1`' is a file name (the default extension '`.res`' being added automatically). The first line of `SOLVE.ARF` must end here, to send `LINEAR` the equivalent of  at the end of the file name. Now if a file `EX1.RES` happens to exist, `LINEAR` will ask us what to do with it, and so the '`R`' makes the choice '`Rewrite`'. If there is no existing file `EX1.RES`, the '`R`' will come in the main menu where '`R`' is not one of the choices, so `LINEAR` will simply ignore it. Next come '`SVV`' for '`Solve`', '`Values`', '`Variables`': the problem is solved, and the values of the variables are listed. The output from these commands is recorded in the file `EX1.RES` (if sent only to the screen, it would simply flash by and be gone without doing much good!). Now `LINEAR` has done all we wanted it to do, so it should exit. This happens automatically at the end of the `.ARF` file.

You will sometimes want to use the **Esc** key in the .ARF file, to return to the main menu. This corresponds to character number 27, which looks like ← when typed. Different editors or word processors may provide different ways (or perhaps no way) to enter this character into a file. It is also called 'Ctrl [': to enter it in SideKick you type **Ctrl P [** (holding the **Ctrl** key down while hitting **P** and **[**), while in EDLIN you use **Ctrl V [**.

## Selecting files

In several commands you are asked to enter the name of a file.

LINEAR uses four types of files. Each has a "default extension":

- .prb for a problem file (a text file expressing the problem in a form that LINEAR can load)
- .dif for a DIF file (used for communicating with spreadsheets, in both directions)
- .tab for a TAB file (an exact image of the state of the problem in memory, may be saved and later loaded)
- .res for a file produced by 'Record' (a text file recording results sent to the screen).

The default extension is added automatically to the name you enter unless you specify an extension. Thus if you enter 'example1' as the name of a problem file, LINEAR will look for EXAMPLE1.PRIB. It is recommended that you use the default extensions for your file names, so that the type of a file is readily recognized, but you can specify any extension you wish. If you want a file with no extension, follow the name with a period, e.g. 'example1.'. If necessary, you can include a drive and/or directory with the file name, e.g. 'c:\linear\examples\example1' (or you could use the 'File Path' command first to change the default drive and/or current directory).

If you are not sure of the exact name of the file you want, or want to check what files there are, you may enter a name containing the 'wild card' characters '?' (which matches any character at that position) or '\*' (which matches any string). Thus 'ex\*' or '??ample?' (to which LINEAR will add the default extension '.prb') will match EXAMPLE1.PRIB but not SAMPLE1.PRIB or EXAMPLE1.TAB, while '\*.\*' will match any file in the current directory. LINEAR will search for all files that match the name you gave, and display them on the screen, as below:

---

```
2 MATCHING FILES
  TEST.PRIB
  TEST2.PRIB
```

---

Up to 63 matching files will be found. Only 22 of these can be shown on the screen at one time, but you can page through them with the **PgDn** key. Use the **↑**, **↓**, **PgUp** and **PgDn** keys to move the menu bar to your choice, and press **↵**. If you want none of the choices given, use the **Esc** key to return to the main menu.

After a file has been chosen, it becomes the "default name" for its type and will be shown in brackets the next time LINEAR asks for a file of that type. To use the default when asked for a file name, just press **↵**. The defaults are initially '\*.prb', '\*.tab' and '\*.dif'.