6 Chapter 6 More sophisticated scripting

6.1 Chapter Overview

We introduce textual entry of solve and plot operations. This is where the operation is specified by keyboard entry alone, without the use of the mouse or Palettes. Textual entry is often preferred by programmers because it is easier to edit scripts written through textual means. We retain the clickable interface for doing quick one-time calculations, or for developing ideas the first time before we start script-writing.

We being to introduce additional concepts in Maple, to enhance what we can solve and plot:

1. In Maple, a character string is a collection of characters delimited by "s: "This is a string." We see how strings are used in the textual entry of labels and colors in plots.

2. Lists e.g. [1,2, x, 3.5] provide a way of organizing multiple results in a single "data container", making it to operate on the whole collection of results while retaining the ability to get at individual results from within the collection. Lists are used in both solve and plot.

3. solve uses two other types of data containers: sequences and sets. We describe how to recognize them and how to extract information from them.

Programmers rely on the on-line documentation to manage the complexity of remembering the details of a knowledge-intensive system such as Maple. They learn/remember how to use a feature by looking up the description, finding an example close to what is desired, and then actively experiment with the example in a fresh worksheet. Reading without experimentation is usually not very productive.

In a previous chapter, we explained assignment and how Maple uses assigned values whenever it sees a name in an expression. We introduce the eval operation, which allows assignments to be done temporarily and immediately forgotten. This can be an attractive alternative if you are concerned about situations where you'd be making many assignments and then undoing them through unassign. eval allows you to evaluate an expression for a particular value of a variable in one line, rather than having to type in the assignment and unassignment as well.

6.2 Textual entry of operations

The textual form of an operation in Maple has the general form:

```
operationName ( sequence of values )
```

The `operationName` can be something like solve or plot. By sequence of values, we mean one or more items, each item separated from the next by a comma. "Sequence" here is the same kind of sequence that was first seen in the previous section on solve (page 56).

Maple will evaluate what you enter in the same way that was described for mathematical expressions possible (page 39). If there are assigned variables mentioned in the sequence, then their values will be used. If Maple knows the `operationName` (e.g. sin, solve, plot), then it will perform the calculation specified by the built-in programming. Otherwise, the result will be more or less what you typed in.

The technical term for the "values" in this situation is actual parameter or argument. Note that the parentheses around the sequence of values are mandatory -- you will either get an error or a result that's far from what you want if you omit the parentheses.

This style of writing things is sometimes called functional notation. In mathematics examples of functional notation are $f(x)$ or $g(3,5)$. In these examples, the name of the operation is the function name $f$ or $g$, while the actual parameters or arguments would be $x$ or the sequence $3,5$.

**Example of textual form of equation solver solve.**
\[
solve(x = 3 \cdot x^2 - 2, x)
\]
\[
\begin{align*}
x &= 3 \cdot x^2 - 2 \\
\text{solve} &
\end{align*}
\]
\[
\{x = -\frac{2}{3}, 1\}
\]

The form of the answer returned by the textual version is slightly different from invoking solve through the clickable interface. The former is typically a form that is easier to work with in scripts.

In this, the \textit{operation Name} (or function name) is \texttt{solve}. There are two arguments. The first argument is the equation \(x = 3 \cdot x^2 - 2\). The second argument is the symbol \(x\).

\[
solve(x = 3 \cdot x^2 - 2, x)
\]
\[
\texttt{solve(x = 3 \cdot x^2 - 2, x)}
\]

What you get if you leave out the parentheses -- not much! Because you left out the parentheses, Maple does not think you are asking for any function to be evaluated, which is how the work gets done.

\[
solve (x = 3 \cdot x^2 - 2, x)
\]

You get another one of those "unable to match delimiters" messages if you forget one of the parentheses.

\[
solve (x = 3 \cdot x^2 - 2, x)
\]

This is what happens if we forget to use the right arrow key to descend from the exponent of \(x^2\). Maple thinks that we are talking about \(x\) to a power called 2-2,x, which understandably it doesn't make sense to it as a power.

### Examples of the textual version of plotting

\[
\text{plot}(x = 3 \cdot x^2 - 2, x = -3..3)
\]

This is a textual form of plot. The first argument is an expression, the second argument is an equation naming a variable and a range of a plot. Note that if we wanted to change the range from -3..3 to -5..2 then we would just edit that line of the worksheet and hit \textit{enter} again. If we wanted to redo the plot in the clickable interface, we would have to right-click and enter all the information all over again.
An attachment at the end of the chapter shows the textual form of common functions, subscripts. These textual forms can be entered from the keyboard wherever the palette entry would work.
6.3 Why are there two different styles for entering operations?

The clickable interface is a good way to get a calculation done quickly, but the actions specified in this way are hard to edit when building scripts. Maple, like most languages, has a textual version of all operations it performs. The editing involved in scripting development can often be easier to do on the textual version. In other words, the clickable interface is good for a one-time calculation, but not so good for the editing and re-execution involved in script reuse.

Another advantage of the textual mode of operation is that the number and variety of operations available in textual form is far greater than what's available in the clickable interface. Maple has several thousand operations. Building a clickable interface to all of them would result in tedious navigation through menus that would either be huge or involve many sublevels.

The downside of using the textual entry is that the developer must spell the text correctly, with the right number and placement of parentheses. Experienced users find that the textual interface is faster to deal with for scripting, while the clickable interface is faster for short, more casual use. Fortunately, in either case one can edit failed attempts and retry, so perfect entry is not necessary to be productive.

Becoming proficient with textual entry of operations is part of the transition technical users make in going from just reuse of other's work to routinely creating their own programming. Without such proficiency, it is hard to realize the full power of the computer in modeling and simulation situations.

6.4 Plotting a list of expressions (multi-plots), plotting lists of numbers (point plots)

Recall that lists in Maple are a way of collecting expressions together into a single object, as discussed in the previous section describing lists and other data containers (page 57). You specify a list by listing the items in the list, enclosed in square brackets [].

If the first argument to plot is a list of expressions, then plot will on a single graph display the plots of all the expressions in the collection. Each one will be displayed in a different color.
Table 6.1: Plotting of multiple expressions

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{plot}({x^2, \sin(x^2)}, x = 0..4))</td>
<td>Plot two expressions on a range where both have comparable-sized results. Here we use the list ([x^2, \sin(x^2)]) to indicate the two expressions that should be plotted.</td>
</tr>
</tbody>
</table>
| Problem: approximately where is the expression \(-3 \cdot x^2 + x - 2\) equal to 5? While we could use solve to tell us exactly, it's often worthwhile to draw a picture and process the situation visually. If we give plot the two expressions \"-5\" and \(-3 \cdot x^2 + x - 2\) then plot will plot not only the parabola (the second expression, it will also plot the expression that is always \(-5\) for any value of \(x\). This corresponds to the horizontal line drawn on the plot. Visually we can see that the parabola is \(-5\) at roughly \(-.8\) and 1.2. We can even get a little more precise by ????.

Plotting multiple expressions simultaneously can be useful when you want to compare them. Assuming that the scales are comparable, one can get a sense of similarity or dissimilarity "at a glance".
We can plot data points rather than smooth curves, if we give the textual form of plot separate lists of $x$ and $y$ coordinates. can be used with the textual version of plot. If we give the textual version of the plot operation two lists of numbers that have the same length, then plot will regard the first list as a list of $x$-coordinates, and the second list as corresponding $y$-coordinates. If you provide plot with the third argument style=point, then it will produce a point plot. Otherwise, it will draw lines connecting each point.

**Table 6.2: Plotting points with lists of numbers**

\[
xList := [1, 2, 3, 4] \quad \text{[1, 2, 3, 4]} \\
yList := [5, 6, 7, -1] \quad \text{[5, 6, 7, -1]} \\
\]

\[\text{plot(xList, yList, style = point)}\]

![Graph of points](image)

We use the textual form of plot to plot the points (1,5), (2,6), (3,7) and (4,-1).
6.5 Strings -- a way to specify titles and labels in plots

A string in Maple is something enclosed in double-quotes: "red", "this is a string?", "Blink++++++182++++++" are all strings. The double-quote symbol is mandatory for a string. Single-quotes ' (also known as apostrophes), backquotes ` (also known as acute accent marks) are not substitutes for double-quotes in writing Maple strings. Characters enclosed by apostrophes or backquotes mean something different to Maple. Use of the wrong punctuation marks will lead to undesired results or error messages.

In addition to being used to input data points in `plot`, lists also can be used in specifying colors and axis labels.

### Example

```
plot(xList, yList)
```

Without the third argument, `plot` will try to connect the points with a curve.

```
plot(xList, yList, Style = point)
```

Error, (in plot) unexpected option: Style = point

Maple cares about whether things are capitalized or not. `Style` is not the same as `style`.

---

**Diagram**

[Image of a plotted graph showing a line connecting points labeled with strings.]
Table 6.3: Plot options and labels

If one of the arguments to the plot operation is of the form
\[ \text{color} = \text{list of color names} \]
then those colors will be used. Most reasonable names will work, but the full list can be seen in on line help
(search for colornames). Note that we are using the textual version of
the symbolic constant \( \pi \). If you can remember how to spell it, it can be
easier than selecting it from the Common Symbols Palette.

Note that because the expressions being plotted are given in a set, the
color assignment is not in the same order that they were typed in. If we
wanted the same order, we should give the plot expressions in a list.

Forgetting brackets for the list of colors.

One of the proficiency issues with textual input is that you have to re-
member all the ( ) parentheses and [ ] brackets. Can you find the the
missing delimiter(s)?

If you leave enough delimiters out, you get error messages that don't
complain about missing delimiters. You have to figure out what the
problem is, which might involve a missing parentheses even if the
message doesn't say so.

The "Error, (in sin)" is a cue that you should look at the places where
you included \( \sin \) in your text and inspect it for problems. It doesn't take
too much effort for you to notice that there's no finishing parentheses
in the first \( \sin(x) \).
One reason why there was no error message about delimiters is that there are multiple missing parentheses. Because there are equal numbers of missing left and right parentheses, there was no alarm for missing delimiters.

\[
\text{plot}\left(\left\{\sin(x), \sin\left(\frac{x}{2}\right), \sin(2\cdot x)\right\}, x=-4\cdot \pi..4\cdot \pi, \\
\text{color} = \text{color} = \left[\text{"Red", "Green", "Blue"}\right]\right)
\]

Error, (in plot) expecting a real constant as range endpoint but received \(-4\pi\)

The problem with this plot is that pi doesn't mean the same thing to Maple as Pi. Maple is case-sensitive. Only Pi means the symbolic math constant having to do with the circumference of a circle.

\[
\text{plot}\left(3.5\cdot x^2 - 2, x=1..5, \text{labels} = \left[\text{"temperature (in degrees C)'}, \\
\text{"pressure in kilopascals'\right]}\right)
\]

If one of the arguments to the plot operation is of the form \(\text{labels} = \text{list of axes titles}\) then those will be used. Each title needs to be a string.

In subsequent work, we will see strings used in other situations within Maple other than for plot titles.

### 6.6 Troubleshooting with strings

The most common mistakes with strings is to leave out the delimiting "s, or to use the wrong kind of delimiters. While the similar-looking keyboard characters ’ (single quote or apostrophe), and ‘ (acute accent or backquote) look like would be equivalent, they are use for other purposes in Maple.

<table>
<thead>
<tr>
<th>What happens when you forget to use the &quot; delimiter in strong, or use the wrong character for the delimiter.</th>
<th>Forgetting to include &quot;s around one of the titles gives a cryptic error message about an &quot;invalid in&quot;. If you were an experienced Maple user, you'd know that in is part of the Maple programming language, and should never be flagged in a string. This would be a clue that there's something wrong around where you entered the first label.</th>
</tr>
</thead>
</table>
| \[
\text{plot}\left(3.5\cdot x^2 - 2, x=1..5, \text{labels} = \left[\text{temperature (in degrees C)}, \\
\text{"pressure in kilopascals'\right]}\right)
\] | \[
\text{plot}\left(3.5\cdot x^2 - 2, x=1..5, \text{labels} = \left[\text{temperature (in degrees C)}, \\
\text{"pressure in kilopascals'\right]}\right)
\] |
The message is not very helpful about telling you how to fix the mistake, though. This unfortunately is typical in most computer programming languages, despite several decades’ effort in building systems software to help people program.

```
plot(3.5*x^2 - 2, x = 1..5, labels = ['temperature (in degrees C)', "pressure in kilopascals"])
```

Error, invalid in

```
plot(3.5*x^2 - 2, x = 1..5, labels
    = ['temperature (in degrees C)',
        "pressure in kilopascals")
```

Putting the wrong kind of quote -- ' instead of " didn't make a string. We got the same indication of a problem as before even though the problem is "wrong kind of quote" rather than "no quote".

### 6.7 Learning through on-line documentation and experimentation

All the options available in the Plot Builder available through the right-click (control-click) interface are also available in the textual version of `plot`. In fact, there are many additional options and varieties of plotting available. The way to find out what the features are and how to invoke them is to consult the on-line documentation.

We can find out more about the textual forms of plotting by invoking Help -> Maple Help and typing `plot` into the search field. When we do so, we see the information in the figure below:
Table 6.4: Plot command help

```
plot - create a two-dimensional plot

Calling Sequence
plot(x, f)
plot(v1, v2)

Parameters
- x: expression in independent variable
- v1, v2: left and right endpoints of horizontal range
- x: independent variable
- v1, v2: x-coordinates and y-coordinates

Description
- The plot(x, f) calling sequence plots the real function f over the horizontal real range from -10 to 10.
- The plot(v1, v2, f) calling sequence plots the real function f over the horizontal real range from v1 to v2.
- The plot(v1, v2) calling sequence creates a curve from the points with x-coordinates v1 and y-coordinates v2, where v1 and v2 are lists or Vectors.

Types of Plots
- For a pictorial listing of the available types of plots, as well as other resources for plotting, see the Plotting Guide. Note that this guide is only available in the Standard interface.

Empty Plots
- If an error occurs during the evaluation of the arguments to the plot command, an empty plot may result.

Interactive Plot Builder
- Maple includes the Interactive Plot Builder, which provides a point-and-click interface to the plotting functionality including two and three-dimensional plots, animations, and interactive plots with sliders.
- To launch the Plot Builder, run the plot[interactive] command. You can also launch the Plot Builder in the Standard Worksheet from the Tools menu. Select Assistants, and then Plot Builder. For more information, see Using the Interactive Plot Builder.

Customizing Plots
- You can customize displayed plots using context menus. To display the context menu for a plot, right-click the plot (for Macintosh, Control-click).
```
We scroll to the bottom of the page and find an example of this. We are looking for a version of plots where v1 and v2 are lists. We don't see something exactly like that but we do see something with Vectors which are similar. Since the document says this should work for lists or vectors, we take the example and see if we can modify it for our own purposes:

Table 6.5: Examples of plot

Evidently, the first list is the values of the x (horizontal) coordinates, and the second list the values of the y (vertical) coordinate. We copy and paste the example into a Maple worksheet and then see if we can get it to work.
According to the documentation, we should be able to get this to work if the first two arguments are lists or vectors. So we edit the example to do lists instead and re-execute the line to see if it works in the same way.
Table 6.7: Modifying the example

To learn about plot options such as colors and labels, we click on the plot,options item under the search results for plot (see green oval in the figure). Clicking on that item produces this information. We see information about color (with another link to see colors), along with possibilities, for labels, symbols, styles, etc. Again, the way to learn the options is through copying and pasting the examples into a fresh worksheet, getting them to work, and then modifying them to suit your own purposes.
Table 6.8: Plot option help page

Plot Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptive</td>
<td>If Adaptive is set to true, the plot will be generated more accurately, but it may take longer to complete. If set to false, the plot will be generated faster but may be less accurate.</td>
</tr>
<tr>
<td>frame</td>
<td>If Frame is set to true, the plot will be displayed in a frame. If set to false, the plot will be displayed without a frame.</td>
</tr>
<tr>
<td>grid</td>
<td>If Grid is set to true, grid lines will be displayed on the plot. If set to false, grid lines will not be displayed.</td>
</tr>
<tr>
<td>axes</td>
<td>If Axes is set to true, the axes will be displayed on the plot. If set to false, the axes will not be displayed.</td>
</tr>
<tr>
<td>legend</td>
<td>If Legend is set to true, a legend will be displayed on the plot. If set to false, a legend will not be displayed.</td>
</tr>
<tr>
<td>title</td>
<td>If Title is set to true, a title will be displayed on the plot. If set to false, a title will not be displayed.</td>
</tr>
<tr>
<td>color</td>
<td>If Color is set to true, the colors of the plot will be displayed. If set to false, the colors of the plot will not be displayed.</td>
</tr>
<tr>
<td>style</td>
<td>If Style is set to true, the style of the plot will be displayed. If set to false, the style of the plot will not be displayed.</td>
</tr>
<tr>
<td>size</td>
<td>If Size is set to true, the size of the plot will be displayed. If set to false, the size of the plot will not be displayed.</td>
</tr>
</tbody>
</table>

Note the "colour". Like Blackberries which are designed just down the road from the Maple company, this is a Canadian product. You will see things like this as well as other indications that it's not an all-American world out there. For example, you can convert pints into Imperial Gallons through Tools -> Assistants -> Units Calculator.

6.8 More operations on lists

So far we have talked only about creating lists, and assigning lists as the value of a variable. You can also generate a sublist of a list, find a particular item in a list by its position index, count the number of items in the list, and convert a list into other types of data.

Table 6.9: Operations on lists

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a list</td>
<td>[ s1 := [a, b, c, a] ]</td>
<td>Lists can contain symbols, numbers, expressions -- anything, even other lists.</td>
</tr>
<tr>
<td></td>
<td>[ [a, b, c, a] ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ s2 := [1, 3.47, 97, -5.9, 2.1] ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ [1, 3.47, 97, -5.9, 2.1] ]</td>
<td></td>
</tr>
<tr>
<td>Specify a sublist of values</td>
<td>[ ts1 := s1[1..3] ]</td>
<td>If a list is followed by another pair of braces with a range inside, then a sublist is computed as a result. Here we have the list that's the first through third items of s1.</td>
</tr>
<tr>
<td></td>
<td>[ ts1 := [a, b, c] ]</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Example</td>
<td>Commentary</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Specify one item from the list</td>
<td>( s1[2] )</td>
<td></td>
</tr>
<tr>
<td>Specify the last item in the list</td>
<td>( s1[-1] )</td>
<td></td>
</tr>
<tr>
<td>Specify a sublist with one item</td>
<td>( s2[3..3] )</td>
<td></td>
</tr>
<tr>
<td>Specify a sublist from the 3rd from the end to the end</td>
<td>( s1[-3..-1] )</td>
<td></td>
</tr>
<tr>
<td>Count the number of items in the list</td>
<td>( n := nops(s2) )</td>
<td>( n := 5 )</td>
</tr>
<tr>
<td>Add together all the items in the list</td>
<td>( \sum_{i=1}^{n} s2[i] )</td>
<td>97.67</td>
</tr>
<tr>
<td>Compute the average of all the numbers in the list.</td>
<td>( \sum_{i=1}^{n} s2[i] ) ( n )</td>
<td>19.53400000</td>
</tr>
<tr>
<td>Convert a list into a sequence</td>
<td>( op(ts1) )</td>
<td></td>
</tr>
<tr>
<td>Convert a list into a string</td>
<td>( convert(s2, string) )</td>
<td></td>
</tr>
</tbody>
</table>

### 6.9 solve, lists and sequences

To solve a system of equations, use a list of expressions or equations for the first argument to `solve`. Use a list of variables as the second argument.

`solve` will return a sequence of lists as the result.

When `solve` finds two solutions for an equation (such as if the equation is quadratic), it will return a sequence of solutions. You can recognize a sequence and distinguish it from a list because; the sequence is missing the enclosing brackets[ ] that a list has.

Part-selection operations work in sequences in a similar fashion as they were described `here (page 86)`.

In `solve`, lists and sequences look as if they are almost interchangable. Later on we will see situations where lists and sequences must be handled differently.
Table 6.10: Solving systems of equations with `solve`

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( eqn := 3x = x^2 - 28 )</td>
<td>We get a sequence of solutions since there is a double root.</td>
</tr>
<tr>
<td>( solns := solve(eqn, x) )</td>
<td></td>
</tr>
<tr>
<td>(-4, 7 )</td>
<td>(6.25)</td>
</tr>
<tr>
<td>( eval(eqn, x = solns[1]) )</td>
<td>We evaluate the equation at the first root and see that it does satisfy the equation.</td>
</tr>
<tr>
<td>(-12 = -12 )</td>
<td>(6.26)</td>
</tr>
<tr>
<td>( eval(eqn, x = solns[2]) )</td>
<td>Same for the second.</td>
</tr>
<tr>
<td>( 21 = 21 )</td>
<td>(6.27)</td>
</tr>
<tr>
<td>( system := [3 \cdot x + 5 \cdot y = 6, 2 \cdot x - 5 = y] )</td>
<td>We want to assign a set as the value of the variable system. Maple tells us that the name is already in use as a built-in function, so it won't let us do that.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>( sys := [3 \cdot x + 5 \cdot y = 6, 2 \cdot x - 5 = y] )</td>
<td>We choose a different variable to assign the set to.</td>
</tr>
<tr>
<td>( [3x + 5y = 6, 2x - 5 = y] )</td>
<td>(6.28)</td>
</tr>
<tr>
<td>( vars := [x, y] )</td>
<td>We specify the set of variables we want to solve for, then call <code>solve</code>.</td>
</tr>
<tr>
<td>( solns := solve(sys, vars) )</td>
<td>We get a list with one element (which itself is a list) as a solution.</td>
</tr>
<tr>
<td>( \begin{bmatrix} x = \frac{31}{13}, y = -\frac{3}{13} \end{bmatrix} )</td>
<td>(6.30)</td>
</tr>
<tr>
<td>( soln1 := solns[1] )</td>
<td>We extract the first element of the list. Notice that that are fewer [ ]s.</td>
</tr>
<tr>
<td>( \begin{bmatrix} x = \frac{31}{13}, y = -\frac{3}{13} \end{bmatrix} )</td>
<td>(6.31)</td>
</tr>
<tr>
<td>( eval(sys, soln1) )</td>
<td>We evaluate the system at the solution that <code>solve</code> has found and verify that this really does satisfy the system of equations.</td>
</tr>
<tr>
<td>( \begin{bmatrix} 6 = 6, -\frac{3}{13} = -\frac{3}{13} \end{bmatrix} )</td>
<td>(6.32)</td>
</tr>
<tr>
<td>( sys2 := [x^2 + y^2 = 25, x + y = 5] )</td>
<td>This system of equations has two distinct solutions, so we get a list with two elements in it. Each element is a distinct solution.</td>
</tr>
<tr>
<td>( [x^2 + y^2 = 25, x + y = 5] )</td>
<td>(6.33)</td>
</tr>
<tr>
<td>( solve(sys2, [x, y]) )</td>
<td></td>
</tr>
<tr>
<td>( [[x = 5, y = 0], [x = 0, y = 5]] )</td>
<td>(6.34)</td>
</tr>
<tr>
<td>( eqn2 := x^2 - 3 \cdot x = 5 )</td>
<td>We can find the non-negative roots of an equation by including the appropriate inequality in the list of relations given to <code>solve</code>. The result by</td>
</tr>
<tr>
<td>( eqn2 := x^2 - 3x = 5 )</td>
<td>(6.35)</td>
</tr>
</tbody>
</table>
\[ \text{solve(eq2)} \]
\[
\frac{3}{2} + \frac{\sqrt{29}}{2}, \quad \frac{3}{2} - \frac{\sqrt{29}}{2} \quad (6.36)
\]

\[ \text{sys3} := [\text{eqn2, } x \geq 0] \]
\[
sys3 := [x^2 - 3x = 5, 0 \leq x] \quad (6.37)
\]

\[ \text{solve(sys3, x)} \]
\[
\left\{ x = \frac{3}{2} + \frac{\sqrt{29}}{2} \right\} \quad (6.38)
\]

\[ \text{solnSet} := \text{solve(sys3)} \]
\[
\text{solnSet} := \left\{ x = \frac{3}{2} + \frac{\sqrt{29}}{2} \right\} \quad (6.39)
\]

\[ \text{solnSet}[1] \]
\[
\quad x = \frac{3}{2} + \frac{\sqrt{29}}{2} \quad (6.40)
\]

### 6.10 Evaluation, eval, and assignment

Suppose that we had an expression relating time \( t \) to voltage registered by a capacitor as it is being charged by a battery. In a mathematics or electrical engineering textbook, we might see this written as

\[ V(t) = 35 + (65 - 35) \cdot \left( 1 - e^{-\frac{t}{3}} \right) \].

We are interested in taking this expression for voltage and doing several calculations with it -- plotting it for a range of \( t \), finding values of \( t \) that correspond to a specified voltage (e.g. "find the time \( t \) when the voltage reached 55 volts"), or finding a voltage corresponding to a specified time (e.g. "find the voltage at \( t=2.5 \) minutes after the start").

If we set up an assignment in Maple \( V := 35 + (65 - 35) \cdot \left( 1 - e^{-\frac{t}{3}} \right) \), then we could calculate the voltage at \( t=2.5 \) minutes by assigning \( t \) the value 2.5 and then evaluating \( V \). The second evaluation will cause the current value of \( t \) to be used. However, if we wanted to plot the expression \( V \) after that, then we'd have problems because whenever we would type \( t \), Maple would use the value of \( t \) rather than the symbol \( t \).

**Evaluating an expression using a particular value of one of the variables in the expression, and then plotting**

\[ V := 35 + (65 - 35) \cdot \left( 1 - e^{-\frac{t}{3}} \right) \]
\[
V := 65 - 30 e^{-\frac{t}{3}} \quad (6.41)
\]

\[ t := 2.5 \]
\[
\quad t := 2.5 \quad (6.42)
\]

\[ V \]
The same problem would happen if we tried to solve an equation involving \( V \) if we had already assigned \( t \) a value.

### Evaluating an expression using a particular value of one of the variables in the expression, and then solving

**restart**

We set up an expression and then evaluate it at \( t=4.7 \) seconds

\[
V := 35 + (65 - 35) \cdot \left(1 - e^{-\frac{t}{3}}\right)
\]

\[
t := 4.7
\]

\[
V
\]

\[
58.73780530
\]

**solve** \((V = 55, t)\)

Warning, solving for expressions other than names or functions is not recommended.

Error, (in solve) a constant is invalid as a variable, 4.7

Because Maple is evaluating the names \( V \) and \( t \) in the solve operation, it is seeing \( \text{solve} \left(35 + (65 - 35) \cdot \left(1 - e^{-\frac{4.7}{3}}\right), 4.7\right) \) which it cannot solve because there are no variables in the equation to solve for.

**unassign('t')**

We can clear the path for solving by unassigning \( t \) first. Doing a restart would not work, because that would also unassign everything, including \( V \). We would lose the expression we want to solve for.
\[ \text{solve}(V = 55, t) \]

\[
3 \ln(3) \quad (6.47)
\]

| \[
\text{It can be tedious to have to remember to unassign variables if we want to go back to using them as symbols in the expression. We recommend using the \textit{eval} operation (also available in the clickable menu as } f(x) \bigg|_{x = a} \text{ instead of assignment, if you are switching back and forth between using values for a variable and using it as a symbol. \textit{eval} returns the same result as if you had done the evaluation, but the evaluation is automatically undone after the calculation is performed.} \]

You can evaluate using values for several variables by giving a list of equations instead of a single equation as the second argument to \textit{eval}.|

| \[
\text{eval}(V, t = 2.5) \]

\[
51.96205374 \quad (6.49)
\]

| \[
\text{solve}(V = 55, t) \]

\[
3 \ln(3) \quad (6.50)
\]

| \[
\text{t} \]

\[
t \quad (6.51)
\]

| \[
\text{Begin parameters} \]

\[
V_i := 35 \quad (6.52)
\]

\[
V_{\text{max}} := 65 \quad (6.53)
\]

\[
v_t := 55 \quad (6.54)
\]

\[
t_0 := 4.7
\]
\[ t_0 := 4.7 \]  
\( \text{(6.55)} \)

**End parameters**

\[ V_{\text{prime}} := V_i + (V_{\text{max}} - V_i) \left( \frac{t}{\tau} \right) - \frac{\tau}{t} \]  
\( V_{\text{prime}} := 65 - 30 e^{-\frac{4.7}{\tau}} \)  
\( \text{(6.56)} \)

\[ \tau_{\text{expr}} := \text{eval}(V_{\text{prime}}, [t = t_0]) \]  
\[ \tau_{\text{expr}} := 65 - 30 e^{-\frac{4.7}{\tau}} \]  
\( \text{(6.57)} \)

\[ \tau_{\text{value}} := \text{solve}(\tau_{\text{expr}} = v_t, \tau) \]  
\[ \tau_{\text{value}} := 4.278124365 \]  
\( \text{(6.58)} \)

\[ t_{\text{expr}} := \text{eval}(V_{\text{prime}}, \tau = \tau_{\text{value}}) \]  
\[ t_{\text{expr}} := 65 - 30 e^{-0.2337472955 t} \]  
\( \text{(6.59)} \)

\[ \text{solve}(t_{\text{expr}} = v_t, t) \]  
\[ 4.699999999 \]  
\( \text{(6.60)} \)

**An example using assignment and eval**

This is an example of how to set up a script using parameters while taking advantage of `eval`. Several symbols in the expression for voltage are set up and assigned as parameters. But we use `eval` to maintain \( \tau \) and \( t \) as symbols in the expression \( V_{\text{prime}} \).

**Begin parameters**

\[ V_i := 35 \]  
\( \text{(6.61)} \)

\[ V_{\text{max}} := 65 \]  
\( \text{(6.62)} \)

\[ v_t := 55 \]  
\( \text{(6.63)} \)

\[ t_0 := 4.7 \]  
\( \text{(6.64)} \)

**End parameters**
\[ V' := V_i + (V_{max} - V_i) \left(1 - e^{-\frac{t}{\tau}}\right) \]

\[ V' := 65 - 30 e^{-\frac{4.7}{\tau}} \quad (6.65) \]

\[ \text{tauExpr} := \text{eval}(V', [t = t0]) \]

\[ \text{tauExpr} := 65 - 30 e^{-0.2337472955 \cdot t} \quad (6.66) \]

\[ \text{tauValue} := \text{solve}(\text{tauExpr} = vt, \text{tau}) \]

\[ \text{tauValue} := 4.278124365 \quad (6.67) \]

\[ tExpr := \text{eval}(V', \text{tau} = \text{tauValue}) \]

\[ tExpr := 65 - 30 e^{-0.2337472955 \cdot t} \quad (6.68) \]

\[ \text{solve}(tExpr = vt, t) \]

\[ 4.699999999 \quad (6.69) \]

### 6.11 Summary of Chapter 6 material

<table>
<thead>
<tr>
<th>Troubleshooting textual input in Maple</th>
<th>Examples with error(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remember to...</td>
<td></td>
</tr>
<tr>
<td>Supply a function name and arguments (parameters)</td>
<td>( \text{solve}x = 3 \cdot x^2 - 2 ) should be ( \text{solve}(x = 3 \cdot x^2 - 2, x) )</td>
</tr>
<tr>
<td>Match delimiters (parenthesis and brackets)</td>
<td>( \text{solve}(x = 3 \cdot x^2 - 2, x) ) should be ( \text{solve}(x = 3 \cdot x^2 - 2, x) )</td>
</tr>
<tr>
<td>Press the right arrow key to exit from variable exponents</td>
<td>( \text{solve}(x = 3 \cdot x^2 - 2, x) ) should be ( \text{solve}(x = 3 \cdot x^2 = 2, x) )</td>
</tr>
<tr>
<td>Set ranges correctly when plotting</td>
<td>( \text{plot}(x - 3 \cdot x^2 - 2, x {-} 3..3) ) should be ( \text{plot}(x - 3 \cdot x^2 - 2, x {-} 3..3) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plotting</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plotting multiple expressions</td>
<td>( \text{plot}([x^2, \sin(x^2)], x = 0..4) )</td>
</tr>
</tbody>
</table>
| Plotting with lists | \[ xList := [1, 2, 3, 4] \]

\[ \text{xList} := [1, 2, 3, 4] \quad (6.70) \]

\[ yList := [5, 6, 7, -1] \]

\[ \text{yList} := [5, 6, 7, -1] \quad (6.71) \]

\( \text{plot}(\text{xList}, \text{yList}, \text{style} = \text{point}) \)
Using multiple colors in a multi-plot

\[
\text{plot}\left( \left\{ \sin(x), \sin\left(\frac{x}{2}\right), \sin(2\cdot x) \right\}, x=-4\cdot \Pi \ldots 4\cdot \Pi, \\
\text{color} = \text{["red", "green", "blue"]} \right)
\]

Set the titles of the axes

\[
\text{plot}(3.5\cdot x^2 - 2, x=1\ldots 5, \text{labels} = \text{["temperature (in degrees C)", "pressure in kilopascals"]})
\]

Using Maple's built-in help
Use Help>Maple Help or press Ctrl-F1 (Command-F1 on a Mac)

Remember that you can click on related topics when viewing the help for a particular command

<table>
<thead>
<tr>
<th>Operations on lists</th>
<th>Examples</th>
</tr>
</thead>
</table>
| Create a list       | \( sl := [a, b, c, a] \) 
|                     | \([a, b, c, a]\) \hspace{1cm} (6.72) |
| Specify a sublist of values | \( ts1 := sl[1..3] \) 
|                     | \([a, b, c]\) \hspace{1cm} (6.73) |
| Specify one item from the list | \( sl[1] \) 
|                     | \( a \) \hspace{1cm} (6.74) |
| Specify a sublist with one item | \( s2[3..3] \) 
|                     | \( [97] \) \hspace{1cm} (6.75) |
| Count the number of items in the list | \( n := \text{nops}(s2) \) 
|                     | \( 5 \) \hspace{1cm} (6.76) |
| Add together all the items in the list | \( \sum_{i=1}^{n} s2[i] \) 
|                     | \( 97.67 \) \hspace{1cm} (6.77) |
| Compute the average of all the numbers in the list. | \( \frac{\sum_{i=1}^{n} s2[i]}{n} \) 
|                     | \( 19.53400000 \) \hspace{1cm} (6.78) |
| Convert a list into a sequence | \( op(ts1) \) 
| Convert a list into a string | \( a, b, c \) \hspace{1cm} (6.79) |
|                     | \( \text{convert}(s2, \text{string}) \) 
|                     | \( \text{"[1, 3.47, 97, -5.9, 2.1"]} \) \hspace{1cm} (6.80) |

Solving a system of equations using lists

Examples
Create the system of equations
\[
\begin{align*}
\text{sys} := & \quad [3 \cdot x + 5 \cdot y = 6, 2 \cdot x - 5 = y] \\
& \quad [3 \cdot x + 5 \cdot y = 6, 2 \cdot x - 5 = y]
\end{align*}
\]
(6.81)

Set the variables of the system
\[
\text{vars} := [x, y]
\]
(6.82)

Solve the system and extract the first solution of possibly many solutions
\[
\text{solns} := \text{solve}(\text{sys}, \text{vars})[1]
\]
\[
\begin{bmatrix}
x = \frac{31}{13}, & y = -\frac{3}{13}
\end{bmatrix}
\]
(6.83)

---

### Evaluating an expression with eval instead of assignment

*eval* allows you to substitute a value for a variable within an expression, without assigning that value to the variable.

- \[
V := 35 + (65 - 35) \cdot \left(1 - e^{-\frac{t}{3}}\right)
\]

- \[
V := 65 - 30 \cdot e^{-\frac{t}{3}}
\]
(6.84)

- \[
eval(V, t = 2.5)
\]
\[
51.96205374
\]
(6.85)

- \[
t
\]
(6.86)