Maple 8 for Calculus at USC[1]

Douglas B. Meade
Department of Mathematics
University of South Carolina
Columbia, SC 29208

Telephone: (803) 777–6183
E-mail: meade@math.sc.edu
Homepage: http://www.math.sc.edu/~meade/

October 14, 2003

[1] This document is continually evolving. The latest version of the document is always available on the WWW at http://www.math.sc.edu/~meade/141L-F03/manual/manual.pdf .
Comments, criticisms, and suggestions for additional — or the removal of — information are encouraged.

Contents

Part I

Using Maple 8 at USC

Chapter 1

Information for Instructors

Teaching a course in which a computer algebra system is available in some form does not have to involve a lot of extra work from the instructor. But, to be effective, the instructor should take a little time to become familiar with Maple's capabilities, features, and limitations. In this section I raise a will attempt to provide some of this essential information. Every instructor should also read Chapter 3 for detailed information about the availability of Maple on the USC campus. In Part II you will find a description of the worksheet interface for Maple 8 (Chapter 6), an Introduction to Maple 8 with an Emphasis on Calculus (Chapter 7), and some additional Tips for Maple Users (Chapter 8).

1.1 Getting Started

Become Acquainted with Student Computing at USC

In Fall 2002, more than 85% of USC students living on campus have access to a personally owned computer with an Internet connection. This means students are not completely dependent on university resources for computer work. However, USC's site license for Maple does not permit students to obtain a copy of the software for their personal use. USC's agreement with Maple allows the software to be installed on any computer owned or leased by the university.[1]

Most students will access Maple in one of the computing labs created by the College of Science and Mathematics. (Students in the College of Engineering and Information Technology can access Maple from the CEIT network.) See Chapter 3 for the specific locations of these labs. Maple is also available in the computer labs located in the dormitories. In the near future Maple is expected to be available in the library computer labs and in the College of Liberal Arts computer labs.

Students do *not* need to have Maple on their personal computer. Students interested in obtaining a copy of Maple should be reassured that the Student Edition is a full version of the software. The Student Edition of Maple can be purchased[2] for $129 from the Maplesoft website

`http://www.maplesoft.com/products/student/` .

There are no limitations, omissions, or other functional differences between the Student Edition and the academic version of the software. (The only differences are the packaging, no manuals, and the user license.)

[1] I believe it is also permissible for faculty to obtain a copy of the software to install on a personally-owned computer *for educational purposes*. People in the Department of Mathematics should contact Minna Moore for a copy of the software. The University contact for Maple is Barbara Koski in CSD.

[2] As Maple is used by more students on the USC campus, it might become possible to to have the Student Edition stocked in local bookstores at a better price.

Maple 8 *vs.* Maple 9

Maple 9 was released in early summer 2003, but Maple 8 is still being used in the labs at USC. The computer labs, and this document, have been prepared specifically for Maple 8. Only minor changes will be needed to update these materials for Maple 9. These changes will be made at the time when Maple 9 will be used in the student computer labs at USC.

If students obtain the Student Edition of Maple 9, they should have very few — if any — problems using the existing materials. The latest information about updates, platforms, and hardware requirements can be found at `http://www.maplesoft.com/products/`.

Web-Based Access to Maple

In addition to the traditional worksheet interface, *maplets* can be used to create customized user interfaces that provide access to specific Maple commands. Maplets can be run either locally (if Maple is loaded) or via the WWW with MapleNet. MapleNet requires Java to create and manage the user interface on the local computer and to establish a connection to Maple running on the MapleNet server. This is a good way for students to access Maple-related content without the expense of obtaining a copy of Maple.[3]

1.2 Computer Labs for Calculus

Weekly Computer Labs

The computer lab components of Calculus I and II are completely planned for you. The students will be learning Maple fundamentals in the context of calculus. Recognize that with one lesson per week, their knowledge will grow slowly. Do not expect them to be independent users until the end of the first semester of Maple usage. If you want to include Maple in your lectures, assign projects, or anything else that involves Maple, please be cognizant of what students have seen and your expectations of their Maple knowledge.

A computer lab experience is prepared for each week of the course. These labs are completely self-contained. Some reinforce topics introduced in lecture, others provide a complete treatment of topics traditionally covered in lecture, and a few labs extend lecture material. Every lab begins with a brief discussion of the purpose of the lab and the necessary mathematical background. New Maple commands and usage is presented in a series of examples. The lab concludes with a series of questions for students to answer. The questions typically require Maple usage similar to that shown in the examples.

The general idea is that the lab assistant will be able present new ideas, including examples, in no more than 15 minutes at the beginning of the lab. This leaves more than 30 minutes for students to work on the questions. Students will have adequate time in the lab period to think about and understand what is required to answer each question. Many students will submit their completed lab at the end of the lab period; the deadline for submitting solutions is about 30 hours after the end of the lab period.

Lab assignments should be prepared as Word documents. It is easy to paste Maple output, including a plot, into Word. The file can then be printed, attached to an e-mail, or submitted through Blackboard.

[3]From a Sun workstation, access to MapleNet requires the use of Netscape 7 (or comparable browser).

The *Information for Students* and *Information for Lab Assistants* Sections are Also for Instructors

Although the information in the next chapters is written for students, you should read it to learn more about the Maple computing environment. If you are already familiar with Maple, you might want to quickly skim through the material. Pay particular attention to facts that you know but are unknown to students or lab assistants. Try to look at this material from the appropriate perspective and recognize how this is different from your perspective on the material. However, if you are a new Maple user, you should read this material more closely, directing particular attention to the terminology and descriptions of manipulations with Maple's graphical user interface.

Chapter 2

Information for Lab Assistants

This section should be written by the current lab assistants, with the input and guidance of George Johnson and Bob Murphy.

Chapter 3

Information for Students

The College of Science and Mathematics (CoSM), College of Engineering and Information Technology (CEIT), and Computer Services Division (CSD) jointly purchase a site license for Maple. This license allows the installation of Maple on any university owned or leased CPU. This is an example of your Student Technology fee being put to use to benefit undergraduate education at USC.

3.1 The CSM Network

Maple is available in every student computer lab on the CSM network maintained in the College of Science and Mathematics. At present these labs are located in LeConte 124, LeConte 303A, EWS 210, and PSC 102. For up-to-date information on instructional computing in the College of Science and Mathematics, please visit the CoSM-IC website at

`http://www.cosm.sc.edu/cosmic/` .

Any student whose major is in the College of Science and Mathematics or currently taking a course in the college is eligible for an account on the CSM network. Speak with your instructor or see information posted in the computer labs for details about obtaining an account on this network.

3.2 The ENGR Network

The general-purpose student computing labs in the College of Engineering and Information Technology are located in Swearingen 1D29 and 3D22. A website with information on Information Technology Services in the CEIT is

`http://www.engr.sc.edu/its/` .

Information about computer accounts on the CEIT network can be obtained from the WWW or the Help Desk in SWGN 1D35.

3.3 Other Networks

Maple is installed on the computers in the university-run computer labs in the dorms. We are currently investigating the possibility of having Maple installed on computers in Cooper Library and in labs maintained by the College of Liberal Arts.

3.4 Blackboard at USC

USC's Blackboard system can be used to present some of the lab materials. Some sections may also use this system to submit and return assignments. In order to successfully use the Blackboard system, including its e-mail and other correspondence features, it is necessary to register your preferref e-mail address (probably not Gamecock E-Mail) and create a Network Username on VIP.

Once you have created a Network Username in VIP, Blackboard can be entered using the URL

`http://blackboard.sc.edu/` .

3.5 Purchasing Maple

You are *not* expected to have Maple on your personal computer. If, however, you would like to purchase a copy of Maple to install on your computer, you should purchase the Student Edition. The Student Edition is available only to students at accredited educational institutions (including USC) but is a full version of Maple. The current price is $129 (compared to $1000 for an academic version and $1800 for a commercial license). Fortunately, the Student Version contains a full version of Maple without any limitations on the size of a problem — another benefit of being a student!

Additional information about the Student Edition, including links to purchase the software online, can be found at

`http://www.maplesoft.com/products/student/` .

3.6 Getting Help with Maple

Your first source of information about a problem you are experiencing with Maple should be Maple's online help system (explained in more detail in Part II of this document – include hyperlink?). Your next sources of assistance should be fellow students and your instructors.

Maple maintains an online Student Center at

`http://www.maple4students.com/` .

This website contains Maple tutorials, lessons for more than 20 courses (ranging from algebra, geometry, and pre-calculus through calculus and differential equations, and beyond including abstract algebra, complex variables, and cryptography), as well as links to many other useful sites.

3.7 Additional Information

A one-page handout with step-by-step instructions for accessing the CSM network, setting up your USC VIP preferences, and preparing to use Blackboard can be found on the web at

`http://www.math.sc.edu/~meade/141L-F03/misc/access.doc` .

An HTML version of this Word document is also available.

A brief Introduction to Maple 8 is provided in Part II of this document. This is a good first source for information about the Maple worksheet interface and an the Maple language. The Maple worksheet from which the latter was created can be obtained from

`http://www.math.sc.edu/~meade/141L-F03/misc/intro-8.mws` ;

an HTML version of this Maple worksheet is also available. A two-page Quick Reference Guide to Maple 8 should be printed (or, at least, bookmarked); the URL is

`http://www.math.sc.edu/~meade/maple/maple-ref.pdf` .

Part II

Lab Assignments for 2003–2004

Chapter 4

Labs for Calculus I

Chapter 5

Labs for Calculus II

Part III

Introduction to Maple 8

Chapter 6

Getting Started

The *Maple worksheet* is the fundamental Maple document. The Maple content for each lab assignments is provided in the form of a Maple worksheet. Because the file extension for a Maple worksheet is `mws`, these ASCII files are sometimes called "`.mws` files". These ASCII files can be used with Maple 8 on any supported operating system[1] — Windows, Linux, Unix, ... The Maple user interface appearance is very similar to other Windows-based applications. Information is entered via the keyboard, by selecting a menu item, or by clicking an icon on one of Maple's menu bars. The purpose of this chapter is to introduce some of the featues, terminology, and essential facts required to make effective use of the Maple worksheet interface.

6.1 Maple Worksheets

A Execution Groups

A Maple worksheet is organized into a series of *execution groups*. Each execution group contains one or more regions. There are four types of regions: input, text, output, and graphics. Only the first two can be explicitly manipulated by the user; the output and graphics regions are created by Maple. Each execution group is identified by a square bracket along the left edge of the Maple worksheet. (See Figure 1.)

An *input region* contains Maple commands, and sometimes includes comments. Input regions are identified by an input prompt ">" and the commands are displayed in red. Commands can be entered either directly from the keyboard, via a palette, or generated by Maple as the result of a *context-sensitive menu*.

A *text region* contains information that is not to be executed by Maple. Typically this includes explanations and mathematical derivations related to the commands in the adjacent input regions. The Maple user interface can be used to perform many document processing operations on text. In addition to changing fonts, sizes, alignment and styles, it is possible to create hyperlinks to help documents, other parts of the current worksheet, or to a document — including another Maple worksheet — on the WWW.

There can be any number of input and text regions, but no more one *output region*, in an execution group. When an execution group has an output region, it appears as the last region in the group. Output generally appears in blue and is "pretty-printed" in standard mathematical format. When errors are detected, appropriate messages are created. These messages appear in pink in an output region.

[1] Maple 9, but not Maple 8, is available for the Macintosh OS.

Maple has commands to create a wide variety of two- and three-dimensional plots. Each graphical object appears in a separate *graphics region* within the output region for that execution group.

Executing Maple Commands

To execute the commands in an execution group use the left mouse button and/or arrow keys to position the cursor anywhere in an input region in the execution group. Then, press the **Return** key. Any output will be presented in an output region at the end of this execution group.

To execute the commands in more than one execution group or to execute all of the commands in the entire worksheet select the appropriate sub-option (either **Selection** or **Worksheet**) from the **Execute** option of the **Edit** menu. All output will be placed in output regions at the end of the execution group from which it was generated.

B Re-executing Maple Commands

Commands in an execution group can be re-executed at any time by placing the cursor in an input region and pressing the **Return** key. If the execution group has an output region, it will be replaced by the output (if any) generated as the commands are re-executed.

Although the result of an assignment appears in an output region, this does not mean that Maple knows about this assignment. Maple only knows the results of commands executed during the current session (or since the last `restart` command). For this reason, when you open a worksheet, you are strongly advised to delete all output from the worksheet before executing any commands.

C Creating New Execution Groups

A new execution group is inserted at the end of a worksheet whenever the last execution group in the worksheet is executed. To insert an execution group elsewhere in the worksheet, use one of the sub-options of the **Execution Group** option of the **Insert** menu or the corresponding icon in the *worksheet bar* (see Figure 6.1).

By default, the new execution group contains a single empty input region. To convert an input region into a text region click on the "capital T" icon found on the worksheet bar. Other conversions between input and text regions can be performed using the first four options in the **Insert** menu.

D Creating, Closing, and Opening Sections

A *section* is a collection of one or more execution groups or sections. The contents of a section are grouped together with a square bracket at the left edge of the worksheet. To distinguish a section from an execution group, the upper corner of the bracket is replaced with a grey box containing either a "+" or "-". When created, a sections is "expanded". Clicking the left mouse button on the "-" in the grey box will "collapse" the section. When collapsed, only the title line of the section is visible.

To create an empty section, select the **Section** option of the **Insert** menu. Execution groups, or other sections, can be added to this section by cutting-and-pasting. A highlighted selection can be inserted into a new section by clicking on the corresponding icon on the worksheet bar.

E Deleting Execution Groups and Sections

To delete an execution group select the entire execution group or section by dragging the mouse or by double-clicking on the square bracket that delimits the object to be deleted. Be sure that the entire group or section is highlighted. Then, either tap the `backspace` or `delete` key on your

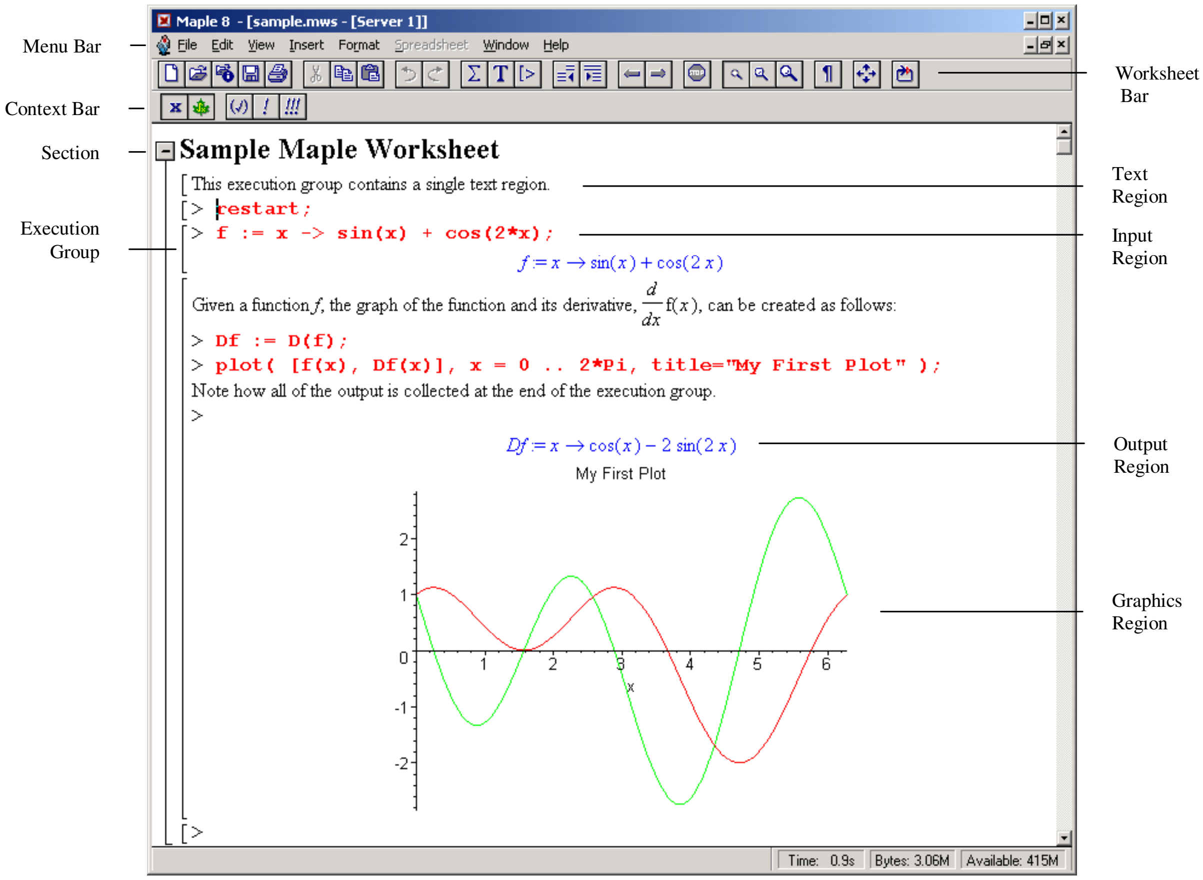


Figure 6.1: The different types of regions and sections in a Maple worksheet.

keyboard, click on the scissors icon on the worksheet bar, or select **Cut** from the **Edit** menu. The same techniques can be used to delete a portion of an execution group except that it is not possible to delete a portion of an output region.

F Entering Maple Commands

The most common method for entering Maple commands is to simply type the commands following the prompt in an input region. Multiple commands can appear within a single input region. If you wish to start subsequent commands on a new line, depress the **Shift** key and press **Return**.

G Context-Sensitive Menus

A *context-sensitive menu* is created whenever a portion of or all of an output region is selected and the right mouse button[2] is clicked (and held down). The items in the context-sensitive menu are operations that Maple has determined to be the most pertinent common operations to perform on the selection. When one of these items is selected, the appropriate command is created in a new execution group immediately following the selection. You can revise the command if necessary and must execute the command to produce the output.

The context-sensitive menus are particularly useful when customizing the appearance of a plot. This menu allows for the creation of a legend, changes to the style and axes, and output to a file in a variety of formats. Note, however, that these changes are lost if the command that created the graph is re-executed.

H Palettes

The *palettes* can be displayed by selecting the appropriate sub-option from the **Palettes** option of the **View** menu. The *symbol palette* contains the Greek letters and the constants e, π, ∞, and i. Clicking on one of these characters produces the corresponding Maple command, *e.g.*, `alpha` for α, `exp(1)` for e, and `Pi` for π. The *matrix palette* can be used to create a template for a matrix with up to four rows and four columns. Each entry of the matrix is initially represented as `%?` and can be replaced by a number or mathematical expression. Note that the `Tab` key can be used to advance to the next `%?`. The *expression palette* creates templates for common mathematical operations ranging from sums and products to roots, exponentials, and trigonometric functions to limits, derivatives, and integrals (both definite and indefinite).

I Saving Worksheets

A Maple worksheet can be saved by either clicking on the diskette icon in the worksheet bar or selecting the **Save** or **Save As ...** options from the **File** menu. Be sure to use unique and descriptive filenames. It is also recommended that you save all Maple worksheets in a small number of folders.

J Getting Help

The **Topic Search** and **Full Text Search** options under the **Help** menu provide two excellent methods for accessing Maple's help information. If you know the keyword for the specific help

[2]This document is prepared under the assumption that the mouse has two or three buttons configured for right-handed use. If your mouse has only one button or has a different mapping for the buttons, you may need to consult the online help and your local support staff to determine the correct translation of the instructions in this document.

document you require, you can use the command, or the shorter name `?`, *e.g.*, `help(plot);` or `?plot`.

The help documents are built from sections and text, input, and output regions. While they are very similar to a Maple worksheet, it is not possible to execute the commands in the **Examples** section. The **Copy Examples** entry of the **Edit** menu copies the entire **Examples** section to the clipboard. If this selection is then pasted into an active worksheet the commands can be executed or modified like any other section.

6.2 Frequently Encountered Problems

A Losing Your Work

Nothing (well, almost nothing) is more frustrating than working for a long period of time and then losing your work as the result of a power outage or system crash. For this reason you should get in the habit of saving your work every few minutes. To activate Maple's AutoSave feature, select **Preferences...** in the **File** menu. In the window that appears, check the **Enable AutoSave** box and enter how often open worksheets should be saved. Note that the backup files are identified by the string `_MAS` after the filename, e.g., `sample_MAS.mws`.

B Syntax Errors

Maple is a programming language that is used to communicate between you, a human, and the computer. Since the computer can only respond to complete and correctly formulated requests, it is essential that you use correct Maple syntax when entering commands. If Maple is unable to make sense of your command, it may produce an error message. (See Figure 6.2.)

Maple is case sensitive. The Maple constant `Pi` is different from the lower- and upper-case Greek letters `pi` and `PI`.

Each Maple command must end with a semi-colon or colon. If this is omitted, or the command is otherwise incomplete, you are likely to see the message "`Warning, premature end of input`". If you look closely, you will also notice that a new input region is inserted in the same execution group. You can type the remainder of the command in the new input region. When you next press **Return**, all of the input regions in the active execution group will be executed.

Another common message is "`Error, wrong number (or type) of parameters in function ...`". When this occurs you will need to check that the command name and parameters are properly entered. The online help worksheets should be consulted for the precise syntax for a command.

C Input Unchanged/Echoed

If you execute the command `feval( sin(1) );`, Maple will respond by displaying the command in an output region. This occurs because Maple thinks `feval` is a user-defined function that the user has not yet specified. Sometimes the cause is a simple misspelling; in this case the correct command would be `evalf( sin(1) );`. Another common source of this problem is when the command is defined as part of a *package* that has not yet been loaded into this session of Maple with the `with` command.

D No Output

In instances when the output is particularly long and messy, you may want to instruct Maple to perform the computation but not to display the results in the workshseet. In these cases you should terminate the command with a colon.

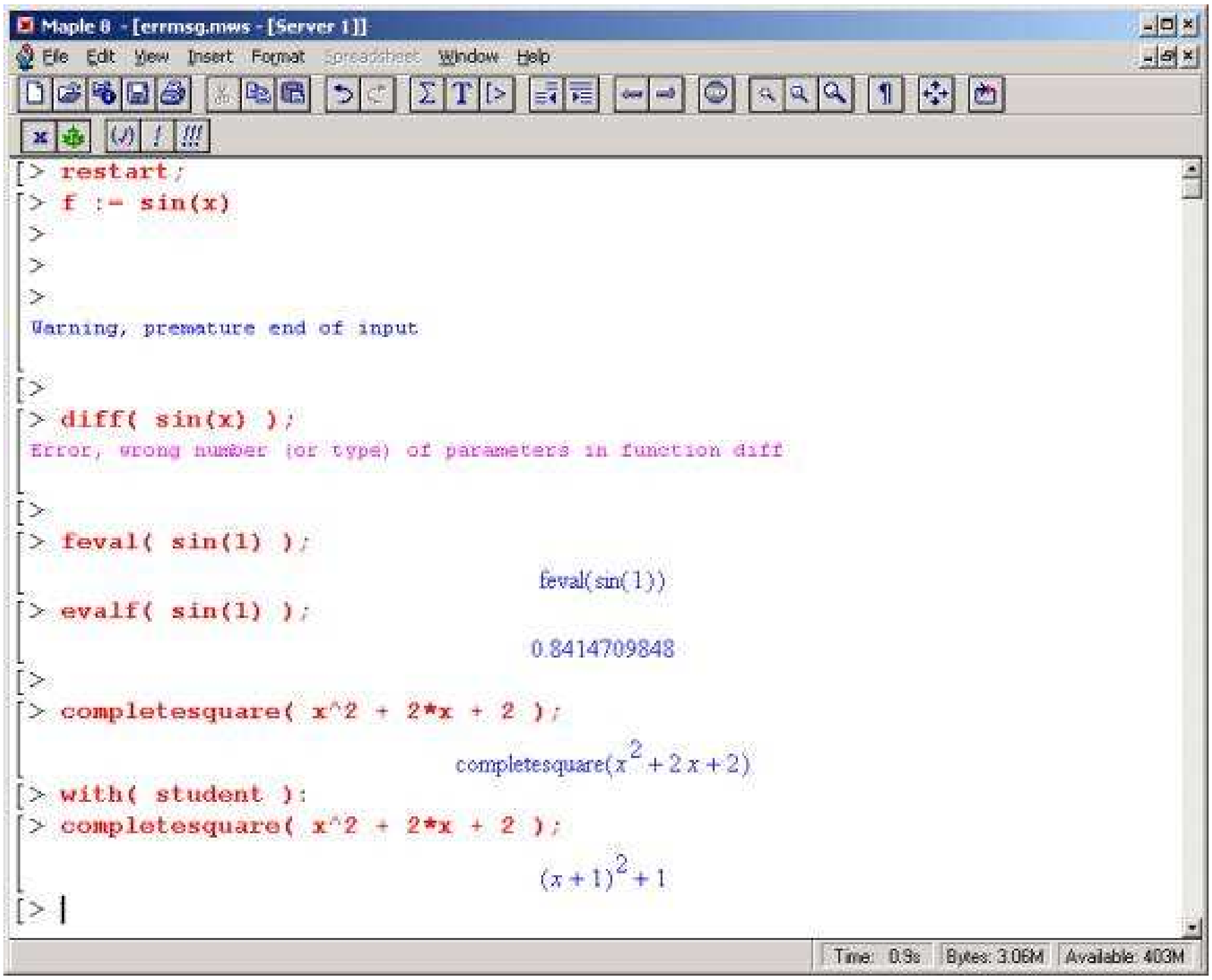


Figure 6.2: Examples of Maple warnings and error messages.

If a command ends with a semi-colon and still does not produce any visible output, then the command has returned `NULL`, the null expression sequence. (`NULL` should not be confused with the empty set, {}.) A NULL response might mean that an equation has no solution or that Maple was unable to find any solution. The online help for the command will generally explain the cases that produce a null response.

E Printing Problems

You have already been advised to save your worksheet prior to printing. This should give you an idea of the importance of this advise.

It is not possible to print only a portion of a worksheet. One way to display only a portion of a worksheet is to use sections and expand only the sections that you wish to see on paper.

ANother solution is to copy (not cut) the regions to be printed to an empty worksheet. To prevent your Maple workspace from becoming too cluttered and fragmented, close — with or without saving — worksheets used for printing.

Bad page breaks are often found in worksheets containing plots. This is because plots cannot be split across pages. The best solution to this problem is to insert physical page breaks (see the **Insert** menu) and manually modify the size of the plot by clicking the left mouse button on the graph and then dragging one of the control points until the graphics region has the appropriate dimensions.

Chapter 7

Calculus and Maple

The best way to become more familiar with Maple is to sit down and use the software. This section is formed from an actual Maple 8 worksheet. The worksheet has been translated to LaTeX using the corresponding **Export As** entry in the **File** menu. The original worksheet is available on the WWW at

`http://www.math.sc.edu/~meade/141L-F03/manual/maple8-overview.mws`.

The HTML version of the same worksheet can be viewed at

`http://www.math.sc.edu/~meade/141L-F03/manual/maple8-overview.html`.

7.1 Introduction

`Maple` is a computer algebra system. While most mathematical software is restricted to working with numerical data, a computer algebra system is designed to perform symbolic manipulations of mathematical expressions. This document presents an overview of Maple with an emphasis on many of the features most likely to be of use in a first-year Calculus course.

7.2 Maple Arithmetic

At the simplest level, you can think of Maple as a powerful calculator that can do symbolic (exact) manipulations as well as floating point (approximate) numerical calculations.

A Addition, Subtraction, Multiplication, and Division

The symbols `+, -, *, and /` are used for addition, subtraction, multiplication, and division, respectively. Don't try any of these on your graphing calculator!

```
> 575754575849849885 + 7489498549859447495989884;
```

$$7489504307405205994488869$$

```
> 87575750 - 4897475988744894574949;
```

$$-4897475988744806999199$$

```
> 6868868686 * 18234987271740;
```

$$125253733060463458733640$$

```
> 99686868612732546598686500000000000 / 5000;
```

$$19937373722546509319737300000000$$

Each command has to end with a `semi-colon` (or `colon`, if you don't wish to see the result).

B Powers

Either `^` or `**` can be used to raise a number to a power.

```
> 55757 ^ 22;
```

$$262172278221307346860617326987246499100441142524856119005736335 03\backslash$$
$$107377146737745572003534597863691126 1049$$

```
> 109 ** 5;
```

$$15386239549$$

Exponentials are handled with the `exp` command.

```
> exp( 2 );
```

$$e^2$$

Euler's constant, e, is obtained as

```
> exp( 1 );
```

$$e$$

The Maple name for ∞ is `infinity`.

```
> infinity;
```

$$\infty$$

C Palettes

Maple has three `palettes` containing shortcuts to entering commands via the keyboard. The **Expression** palette can be used to compute powers, roots, elementary transcendental functions, limits, derivatives, and other basic calculus-based quantities. To open this palette, pull down the **View** menu, choose **Palettes** and then select **Expression Palette**. Drag the palette to a position where it does not interfere with the current worksheet. (You may also need to resize the Maple and/or worksheet window.)

To use the palette to enter a quantity like $\sqrt{390625}$, position the cursor in an empty input region,

```
> sqrt(%?);
```

$$\sqrt{?}$$

click on the symbol $\sqrt{a}$ in the palette. This produces `sqrt( %?)` with the argument (`%?`) selected. Type `390625` and then execute the group. Your final input and output should appear as

```
> sqrt(390625);
```

$$625$$

When a palette generates a template that involves more than one argument, the **Tab** key can be used to move from one argument to the next. For example, to compute $\frac{757555}{5}$, use the a / b button on the palette, enter 757555, press **Tab**, enter 5, then press **Return** to obtain

```
> ((757555)/5);
```

$$151511$$

D Exact vs. Approximate Calculations

Maple is designed to provide exact answers to mathematical computations.

```
> sqrt( 27 );
```

$$3\sqrt{3}$$

While the exact simplification in the previous example is useful, there are times when an exact answer is not helpful. For example, the following rational number is returned unchanged because it is already in reduced form.

```
> 5899 / 7;
```

$$\frac{5899}{7}$$

There are several ways to instruct Maple to produce an approximate value for this number. Maple will display an answer as a floating point number if at least one number in the calculation is a floating point number (i.e., contains a decimal point).

```
> 5899.  / 7;
```

$$842.7142857$$

The `evalf` command can also be used to force Maple to **__eval__**uate using **__f__**loating point arithmetic

```
> evalf( 5899 / 7 );
```

$$842.7142857$$

By default, Maple performs all floating point computations using 10 significant digits. The following command performs the same calculation except that only 5 significant digits are used in the calculations. (See also the online help for `Digits`.)

```
> evalf[5]( 5899 / 7 );
```

$$842.71$$

7.3 Assigning Values to Names

A Using Previous Results

The *ditto operator* `%` always represents the result of the last command executed by Maple.

```
> 625 / 125;
```

$$5$$

At this point, the most recent result computed is 5. This can be squared with the command

```
> % ^ 2;
```

$$25$$

It is permissible to include more than one command in a single input region. For example,

```
> sqrt( 23.1 );
  2 - 9;
```

$$4.806245936$$
$$-7$$

In this case, the most recent result is -7. The next most recent result can recalled with `%%`.

```
> %%` ^ 2 + %;


Warning, incomplete quoted name;  use ` to end the name
```

The ditto operators should be used sparingly. You should get in the habit of assigning results to explicit names.

B Assignments

The Maple command for assigning a value to a name is the two-character sequence " `:=` ". The single character " `=` " is used to form equations or to test for equality of two objects. Names generally consist of a letter followed by one or more letters, numbers, and underscores.

The commands to assign `x` the value 2 and `y` the value 3 are

```
> x := 2;
```

$$x := 2$$

```
> y := 3;
```

$$y := 3$$

The name `prod` will be assigned the product of `x` and `y`

```
> prod := x * y;
```

$$prod := 6$$

From now on, the name `prod` will be replaced with this value. Thus,

```
> prod;
```

$$6$$

and, if the value of `x` is changed, the value of `prod` is not affected

```
> x := 9;
  prod;
```

$$x := 9$$
$$6$$

The `unassign` command removes assignments to the listed names; note that the single quotes are required to prevent Maple from evaluating these names to their assigned values. (To erase *all* assignments, it is easier to use `restart`.)

```
> unassign( 'x', 'y', 'prod' );
> x, y, prod;
```

$$x,\ y,\ prod$$

Note that a different behavior is obtained if we define `prod` as before, but before assigning values to `x` and `y`,

```
> prod := x * y;
```

$$prod := x\,y$$

When values are assigned to `x` and `y`, these values are used to compute the current value of `prod`.

```
> x := 10;
  y :=  9;
  prod;
```

$$x := 10$$
$$y := 9$$
$$90$$

And, if one or both of `x` and `y` are changed, the value of `prod` changes as well.

```
> x := 3;
  prod;
```

$$x := 3$$
$$27$$

The difference in these two examples is whether the names used in the definition of `prod` have values *at the time the assignment is made.*

C Suppressing Output

The last few examples have had more than one command in each input region. In this example, multiple names receive values in a single assignment. In this case the result of this assignment is suppressed because the assignment command is terminated with a colon.

```
> x, y := 90, 30:
  x; y; prod;
```

$$90$$
$$30$$
$$2700$$

7.4 Maple Commands

```
> restart;
```

A Built-In Commands and Constants

Maple commands consist of a string of letters (and numbers) followed by one or more arguments enclosed in round brackets (parentheses). The `evalf` and `unassign` commands have already been encountered in this worksheet. Here are a few more examples.

If m and n are integers, `$` m **..** n returns the list of all integers from m to n inclusive.

```
> nums := $ 1 ..  10;
```

$$nums := 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$$

If *exprseq* represents a expression sequence of numbers, then `min(` *exprseq* `)` will return the smallest number in *exprseq*.

```
> min( nums );
```

$$1$$

To find the smallest number in a set or list, the surrounding brackets must be eliminated to obtain an expression sequence.

```
> L := [ 2, 3, 6 ];
```

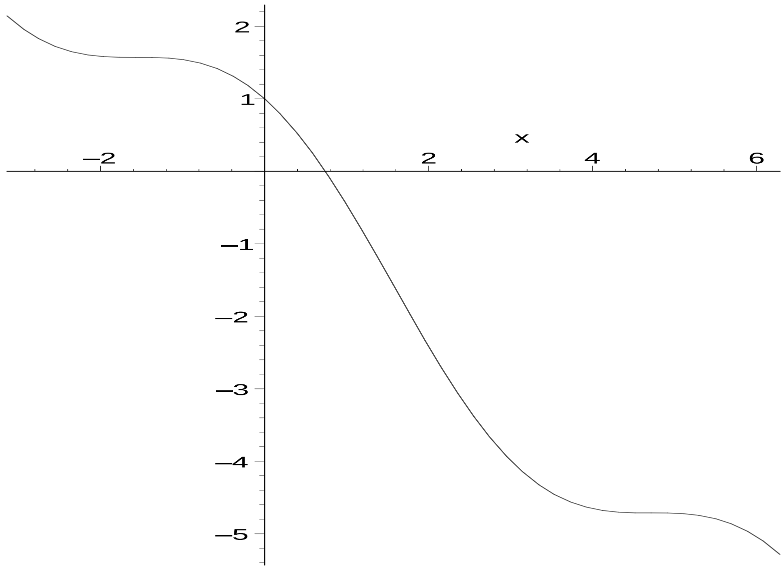
$$L := [2, 3, 6]$$

```
> min( L );
Error, (in simpl/min) arguments must be of type algebraic
> min( L[] );
```

$$2$$

The following `plot` command will plot the function $f(x) = \cos(x) - x$ on the domain $[\ -\pi\ ,\ 2\,\pi\ ]$.

```
> plot( cos(x) - x, x = -Pi ..  2*Pi );
```



In Maple, the equation $\cos(x) - x = 0$ is represented exactly as we would write it by hand. Note the use of `=` to form an equation, not the assignment operator `:=` . The following `fsolve` command locates a solution to $\cos(x) - x = 0$ near $x = 1$.

```
> fsolve( cos(x) - x = 0, x = 0 ..  2 );
```

$$0.7390851332$$

As mentioned previously, some Maple names are predefined to standard constants. For example, `Pi` is π. Recall that Euler's constant, e, is obtained with `exp(1)`.

```
> evalf( Pi );
  evalf( exp(1) );
```

$$3.141592654$$
$$2.718281828$$

```
> evalf( Pi^exp(1) ) < evalf( exp(Pi) );
```

$$22.45915771 < 23.14069264$$

The command for the square root of a number x is `sqrt(x)`.

```
> sqrt( 4 );
```

$$2$$

Note that Maple has no trouble handling the square root of a negative number; `I` is the imaginary unit, i.e., $I^2 = -1$.

```
> sqrt( -4 );
```

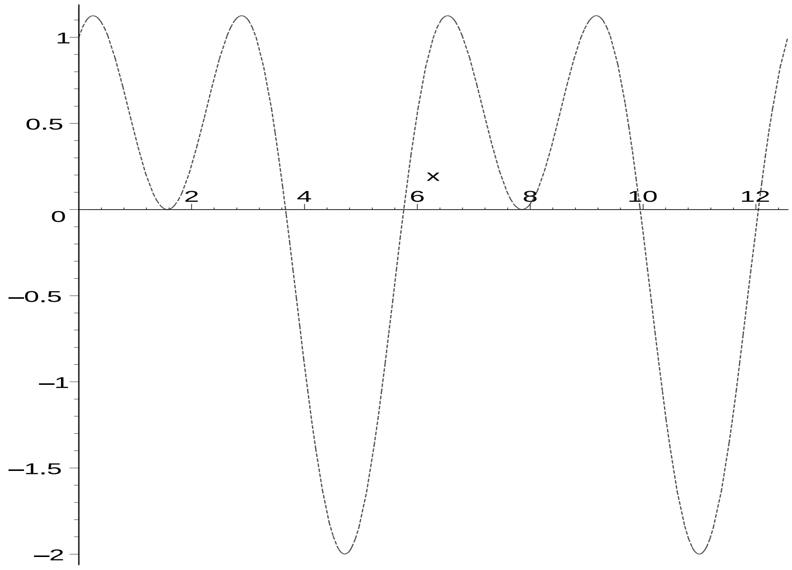
$$2\,I$$

Each of these quantities could also have been assembled using the expresssion palette.

B Command Options

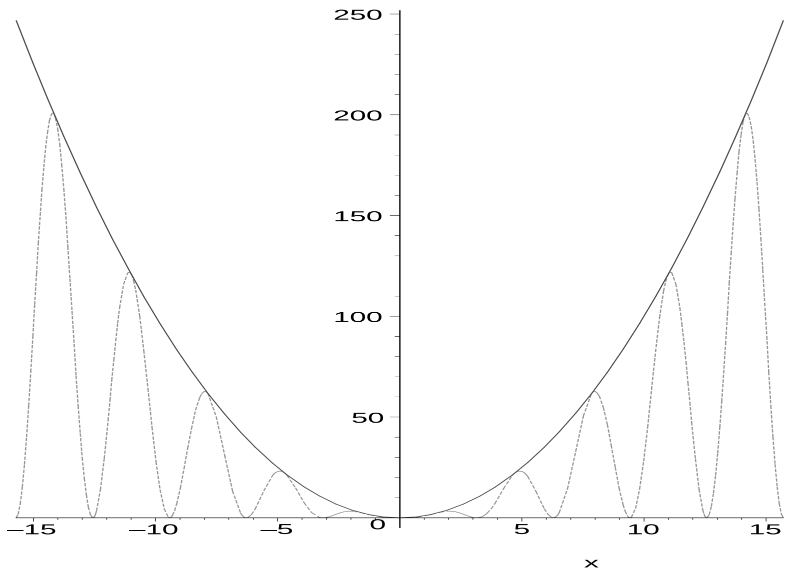
Many Maple commands, particularly the plotting commands, accept optional arguments for customizing the output. For example, the option `linestyle=3` plots the command using a dashed line.

```
> plot( sin(x) + cos(2*x), x = 0 ..  4*Pi, linestyle = 3 );
```



In the next `plot` command, two functions, x^2 and $x^2 \sin(x)^2$, are plotted simultaneously with the first function appearing as a dashed line and the second as a solid line. (Although it is not apparent in a hardcopy of this worksheet, Maple displays the first plot in red and the second in green. The `color=` option can be used to control the colors used in a plot.)

```
> plot( [x^2, x^2*sin(x)^2], x = -5*Pi ..  5*Pi, linestyle = [ 1, 3 ] );
```



For a complete listing of the optional arguments for the `plot` command, see the online help for `plot,options`.

C Online Help

The Maple command for accessing information in the online help database is `help(` *keyword* `);` , or the abbreviated form `?` *keyword*. The help information appears in a separate window within Maple. While the help document appears to be very similar to a Maple worksheet, it is not possible to execute any commands that appear in a help document. Commands in the Examples section of a help worksheet can be copied individually or in their entirety and pasted in worksheet. To return to an active worksheet, either close the help window or select the desired window in the list of windows under the **Window** menu.

```
> help( fsolve );
> ?plot,color
```

The **Help** menu can also be used to browse the online help database. This is particularly powerful when you do not know the command name. The `New User's Tour` is particularly useful for new users.

D Packages

In addition to the standard Maple functions available to you at the beginning of every Maple session, there are an ever-growing number of additional functions contained in *packages* that must be loaded into the Maple session prior to their use. The `with` command is used to load a package. Two of the more common packages for use in a Calculus course are the `plots` and `Student[Calculus1]` packages.

```
> with( plots );
```

```
Warning, the name changecoords has been redefined
```

[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *graphplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *interactive*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *odeplot*, *pareto*, *plotcompare*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedra_supported*, *polyhedraplot*, *replot*, *rootlocus*, *semilogplot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

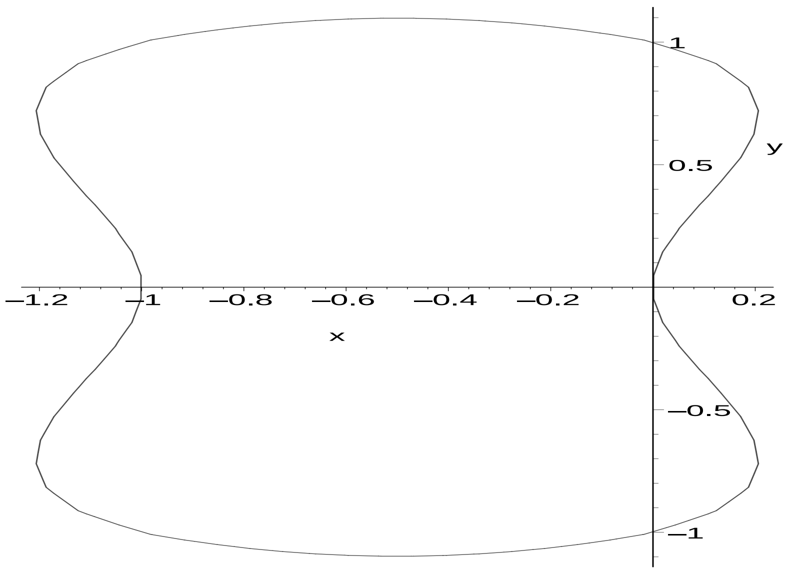
```
> with( Student[Calculus1] );
```

[*AntiderivativePlot*, *ApproximateInt*, *ArcLength*, *Asymptotes*, *Clear*, *CriticalPoints*, *DerivativePlot*, *ExtremePoints*, *FunctionAverage*, *FunctionChart*, *GetMessage*, *GetNumProblems*, *GetProblem*, *Hint*, *InflectionPoints*, *Integrand*, *InversePlot*, *MeanValueTheorem*, *NewtonQuotient*, *NewtonsMethod*, *PointInterpolation*, *RiemannSum*, *RollesTheorem*, *Roots*, *Rule*, *Show*, *ShowIncomplete*, *ShowSteps*, *Summand*, *SurfaceOfRevolution*, *Tangent*, *TaylorApproximation*, *Understand*, *Undo*, *VolumeOfRevolution*, *WhatProblem*]

The output of a successful `with` command is the list of commands that have been added to Maple's repertoire. If this list is unwanted, use a colon to terminate the command.

One of the commands that is now defined is `implicitplot`.

```
> implicitplot( x^2 + y^4 = y^2 - x,
                x = -1.3 ..  0.3, y = -1.2 ..  1.2,
                scaling=constrained );
```



The `scaling=constrained` option instructs Maple to use the same scaling for each axis in the plot.

E Creating Functions

Users can create their own Maple commands, including Maple implementations of mathematical functions. For example, the function $\mathrm{f}(x) = x^2 \sin(x)^2$ can be defined using

```
> f := x -> x^2 * sin(x)^2;
```

$$f := x \to x^2 \sin(x)^2$$

The variable x is a dummy variable; it is replaced by whatever object appears as the first argument to **f**.

```
> f( y );
```

$$y^2 \sin(y)^2$$

```
> f( fred );
```

$$\mathit{fred}^2 \sin(\mathit{fred})^2$$

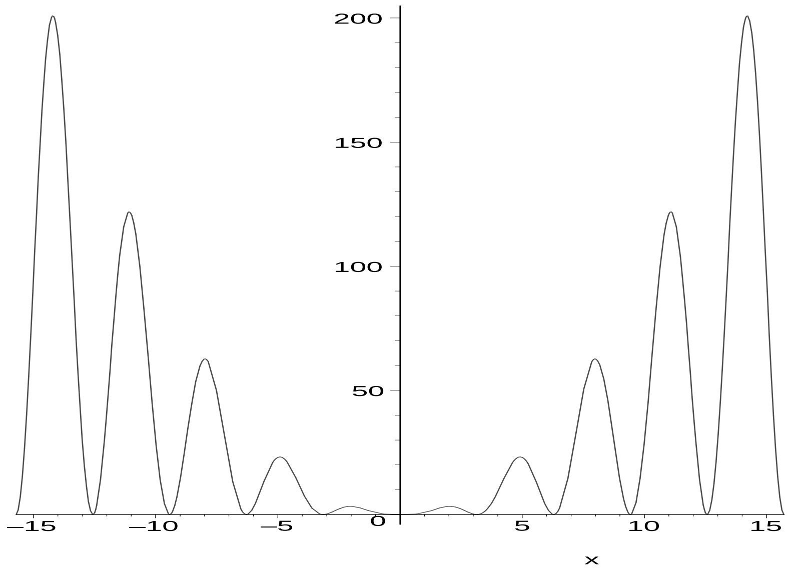
```
> f( Pi/2 );
```

$$\frac{\pi^2}{4}$$

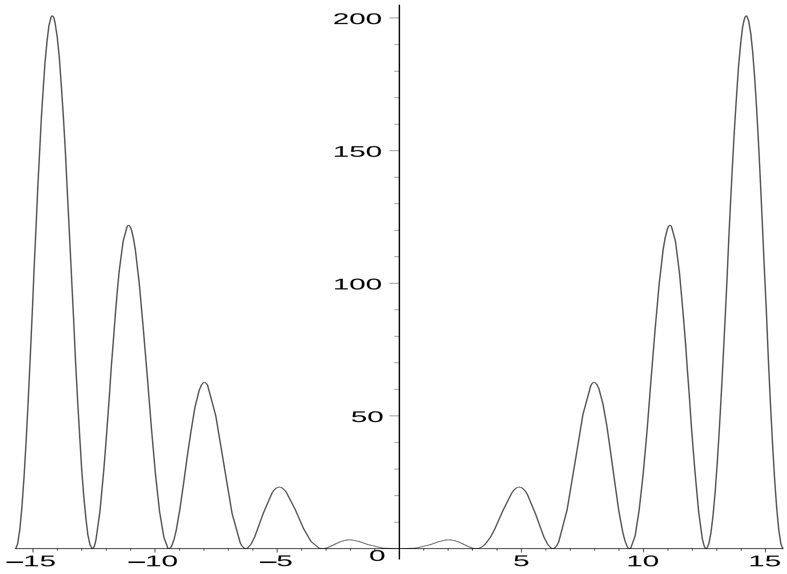
Notice how Maple automatically simplifies the value of the function when possible.

To plot the function you can use either of the following commands

```
> plot( f(x), x = -5*Pi ..  5*Pi );
```



```
> plot( f, -5*Pi .. 5*Pi );
```



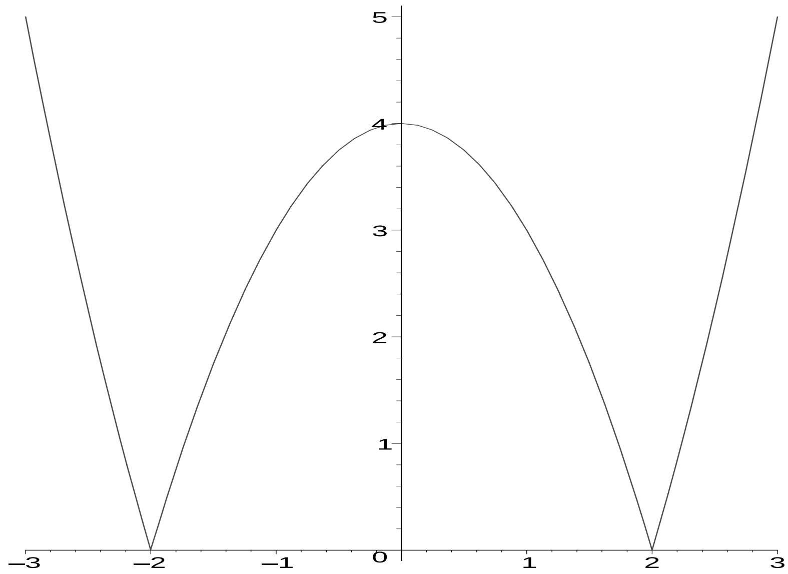
The plots are identical, except for the label on the horizontal axis.

The `unapply` command provides a second way to define a Maple function.

```
> g := unapply( abs(x^2-4), x );
```

$$g := x \to |x^2 - 4|$$

```
> plot( g, -3 .. 3 );
```



Notice that the output of the `unapply` command uses the same arrow notation as was used to define `f` above. The main difference between the arrow operator and `unapply` is that the argument to `unapply` is evaluated *before* the function is created. In contrast, the right-hand side of the arrow operator is *not evaluated.*

7.5 Lists and Sets

A Definitions

Two of the most important data structures in Maple are the list and set. A *list* is an ordered expression sequence contained in square brackets, `[` *exprseq* `]`, and a *set* is an unordered expression sequence contained in braces, { *exprseq* }. (An *expression sequence* is a comma-separated collection of numbers, names, equations, or other Maple objects.)

```
> S := { 1, 3, 111 };
```

$$S := \{1, 3, 111\}$$

```
> L := [ $ 6 .. 10 ];
```

$$L := [6, 7, 8, 9, 10]$$

Notice that the elements of a list or set can be any valid Maple object, including another list or set.

```
> SS := { L, S };
```

$$SS := \{\{1, 3, 111\}, [6, 7, 8, 9, 10]\}$$

```
> LL := [ L, S ];
```

$$LL := [[6, 7, 8, 9, 10], \{1, 3, 111\}]$$

Look carefully at the previous results. Even though the only difference in the definition of `LL` and `SS` is the type of brackets, the order of the elements in `SS` might appear in the opposite of the order in which they appeared in the definition. Recalling that the elements of a set are not ordered, this is not surprising. (It should also not be surprising to know that the order in which Maple displays the elements of a set can change from one session to another.)

B Creating Lists and Sets

Except for the surrounding brackets, lists and sets are created in exactly the same ways. We have already seen how to create lists and sets from explicit collections of numbers and with the repetition operator (`$`). The `seq` and `map` command provide two additional methods for creating lists and sets.

The `seq` command generates an expression sequence consisting of terms formed from the first argument for each value of the second argument. For example, the squares of the first ten positive integers is

```
> pts := [ seq( i^2, i = 1 .. 10 ) ];
```

$$pts := [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]$$

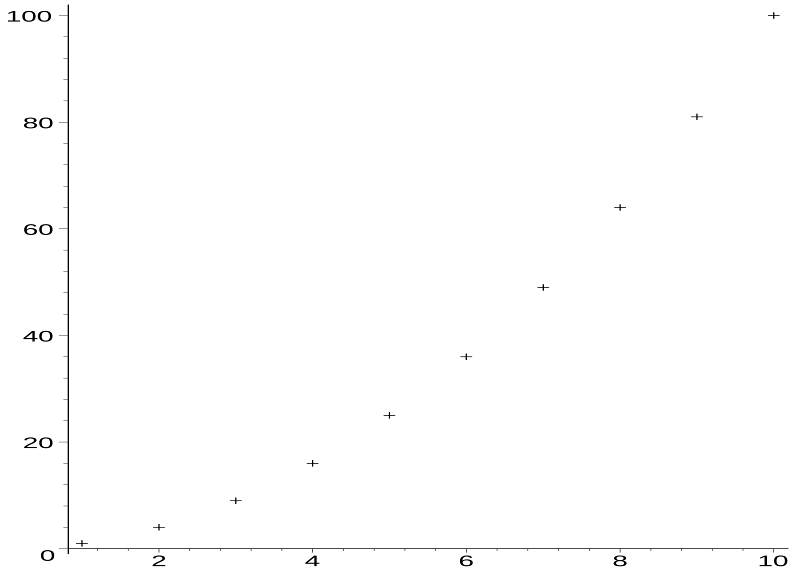
This list could also be created using `$` as

```
> [ i^2 $ i = 1 .. 10 ];
```

$$[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]$$

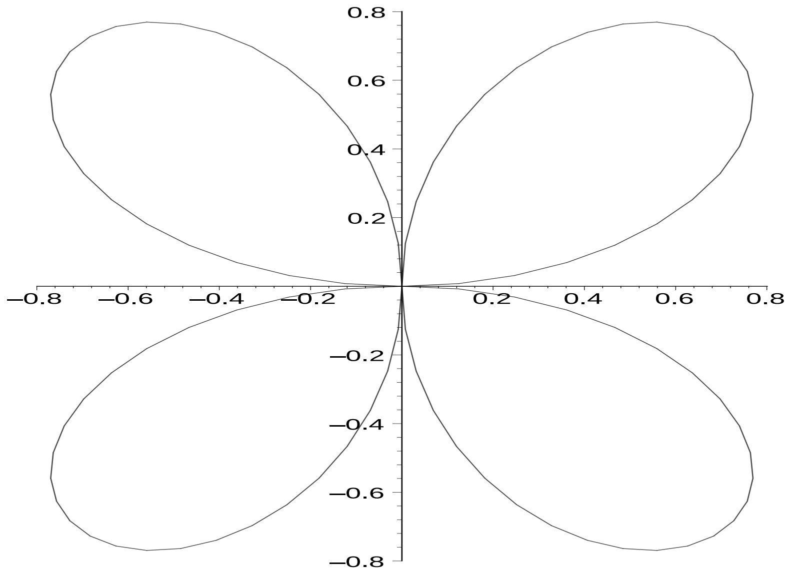
To plot a list of *values* against their index in the set, use the `listplot` command from the `plots` package. The `style=point` optional argument instructs Maple to display discrete points (not connected), `symbol=cross` sets the plot symbol (the default is a diamond), and `symbolsize=20` sests the size of the symbols (in points; the default is 10).

```
> listplot( pts, style=point, symbol=cross, symbolsize=20);
```



A list of 101 *points* on the polar curve $r = \sin(2\,t)$ is constructed and displayed next. The list is not displayed as it is quite lengthy. Because the list elements include both the x- and y-coordinates, the `plot` command can be used.

```
> r := t -> sin(2*t):
> rose4 := [ seq( [r(t*Pi/50)*cos(t*Pi/50), r(t*Pi/50)*sin(t*Pi/50)],
                  t = 0..100 ) ]:
> plot( rose4 );
```



The same plot could also have been created directly with any of the following variations of the `plot` command.

```
> listplot( rose4 ):
> plot( [ r(t) * cos(t), r(t) * sin(t), t = 0 .. 2*Pi ] ):
> plot( r(t), t = 0 .. 2*Pi, coords=polar ):
```

The last two commands show to create the graph of a parametric curve (C: $\mathrm{x}(t) = \mathrm{r}(t)\cos(t)$, $\mathrm{y}(t) = \mathrm{r}(t)\sin(t)$ for t in $[0, 2\pi]$) and how to plot a polar function by specifying only the radius function and the range for the polar angle.

C Extracting Elements of a List or Set

Much more than plotting can be done with a list or set. The third element of the list `pts` can be accessed as

```
> pts[3];
```

$$9$$

Notice that it does not make sense to talk about the third element of a set. While Maple will not object to this, you should not expect to receive the same result every time the command is executed. The `select` and `remove` commands are designed to extract elements of a set that meet certain criteria. For example, the subset of `S` containing all elements that are in the open interval (-10, 10) can be found as follows. (The `evalb` command returns either *true* or *false* (or *FAIL*) indicating the Boolean value of its argument.)

```
> S;
```

$$\{1, 3, 111\}$$

```
> select( x -> evalb( abs( x - 10 ) < 20 ), S );
```

$$\{1, 3\}$$

The number of elements in a list or set can be determined with the `nops` command.

```
> nops( rose4 );
```

$$101$$

Recall that each element of `rose4` is an ordered pair – actually, a two-element list.

```
> rose4[ 10 ];
```

$$[\sin(\frac{9\,\pi}{25})\cos(\frac{9\,\pi}{50}),\ \sin(\frac{9\,\pi}{25})\sin(\frac{9\,\pi}{50})]$$

A floating-point approximation to this point is

```
> evalf( % );
```

$$[0.7639707480, 0.4848305795]$$

The y-coordinate of the tenth element of the list is

```
> rose4[ 10, 2 ];
```

$$\sin(\frac{9\,\pi}{25})\sin(\frac{9\,\pi}{50})$$

Negative indices can be used to reference elements relative to the end of the list. The last point in `rose4` is

```
> rose4[ -1 ] = rose4[ nops(rose4) ];
```

$$[0, 0] = [0, 0]$$

The first ten elements of rose for can be specified using `rose4[ 1 .. 10 ]`. The floating point approximations to the x-coordinates of each of these points can be obtained with

```
> seq( evalf(p[1]), p = rose4[ 1 .. 10 ] );
```

0., 0.1250859171, 0.2467288932, 0.3616040548, 0.4666184965, 0.5590169945, 0.6364758026, 0.6971812263, 0.7398902011, 0.7639707480

Sets can also be manipulated using the standard set operators: `union`, `intersect`, and `minus`. Each of these commands can be used both as an infix operator, following standard mathematical notation, or as a prefix operator, which looks more program-like.

```
> word := { "to", "too", "two" };
  num := { "one", "two", "three", "four" };
```

$$word := \{\text{“to”, “too”, “two”}\}$$

$$num := \{\text{“two”, “one”, “three”, “four”}\}$$

```
> word union num;
```

$$\{\text{“to”, “too”, “two”, “one”, “three”, “four”}\}$$

```
> `union`(word, num);
```

$$\{\text{“to”, “too”, “two”, “one”, “three”, “four”}\}$$

```
> word intersect num;
```

$$\{\text{“two”}\}$$

```
> num minus word;
```

$$\{\text{“one”, “three”, “four”}\}$$

The quotes on the command in the prefix form of the commands are required (to delay evaluation). The prefix forms of `union` and `intersect` can accept any number of sets.

D Extracting Solutions to an Equation

When Maple finds the solution to an equation or system of equations, the output will be displayed as an expression sequence. You will often want to convert this to a list or set by inserting the appropriate brackets around the `solve` (or `fsolve`) command.

```
> sol := [ fsolve( x^2 = 3, {x} ) ];
```

$$sol := [\{x = -1.732050808\}, \{x = 1.732050808\}]$$

```
> sol[ 1 ];
```

$$\{x = -1.732050808\}$$

Approximations to the four solutions to the quartic polynomial equation $x^4 - 3\,x^2 + x = 0$ can be found, as a set, using

```
> realsol := { fsolve( x^4 - 3*x^2 + x = 0, x ) };
```

$$realsol := \{-1.879385242, 1.532088886, 0.3472963553, 0.\}$$

In general, `fsolve` returns at most one (real) solution. For a polynomial, all real valued solutions are returned. To obtain complex-valued solution, add `complex` as an optional argument.

```
> complexsol := { fsolve( x^4 - 3*x^2 + x = 4, x, complex ) };
```

$$complexsol := \{1.893967041, 0.1004849083 + 0.9990160212\,I, -2.094936857,$$
$$0.1004849083 - 0.9990160212\,I\}$$

Any individual solution can be obtained as above, but the result would depend on the order of the elements in a set – which is unpredicable.

```
> realsol[3];
```

$$0.3472963553$$

The largest element of a set can be found using

```
> max( realsol[] );
```

$$1.532088886$$

The absolute value of each solution can be obtained using

```
> map( abs, complexsol );
```

$$\{1.893967041, 2.094936857, 1.004056885\}$$

To sort the four real roots in increasing order, re-express the solutions as a list and then call the `sort` command

```
> realsol2 := convert( realsol, list );
```

$$realsol2 := [-1.879385242, 1.532088886, 0.3472963553, 0.]$$

```
> sort( realsol2 );
```

$$[-1.879385242, 0., 0.3472963553, 1.532088886]$$

To sort the solutions by other orderings, e.g., the magnitude of the solutions, a user-defined ordering can be specified as the second argument to the `sort` command

```
> sort( realsol2, (a,b) -> evalb( abs(a) < abs(b) ) );
```

$$[0., 0.3472963553, 1.532088886, -1.879385242]$$

To conclude this discussion, consider the system of equations that describes the set of all points on the unit circle, $x^2 + y^2 = 1$, and the line, $x + 2\,y = 1$.

```
> eq1 := x^2 + y^2 = 1;
  eq2 := x + 2*y = 1;
```

$$eq1 := x^2 + y^2 = 1$$
$$eq2 := x + 2\,y = 1$$

The system of equations and variables are each specified as sets that, when fed to the `solve` command, show two solutions

```
> syssol := [ solve( { eq1, eq2 }, { x, y } ) ];
```

$$syssol := [\{x = 1,\, y = 0\},\, \{y = \frac{4}{5},\, x = \frac{-3}{5}\}]$$

Each solution is a set of equations giving the x- and y-coordinates of a point on both curves. Note that the order of the equations within each solution may not be consistent. These solutions can be converted to ordered pairs using

```
> syspts := seq( eval( [x,y], pt ), pt = syssol );
```

$$syspts := [1, 0],\, [\frac{-3}{5}, \frac{4}{5}]$$

The two curves involved in this problem can be plotted with the `implicitplot` command from the `plots` package. The two solutions can be plotted using `plot` with `style=point`. The `display` command (also from the `plots` package) can be used to display the information in both plots in a single plot. The output from the plot-creating commands in the definition of `p1` and `p2` is the corresponding “plot data structure”, not a graphical object. (If you really want to see this, change the colons to semi-colons.)

```
> p1 := implicitplot( { eq1, eq2 }, x = -2 ..  2, y = -2 ..  2,
                      scaling=constrained ):
  p2 := plot( [ syspts ], style=point, color=black, symbol=circle,
              symbolsize=30 ):
  display( [ p1, p2 ] );
```

Note that the `implicitplot` command cannot accept a list as its first argument. This inconsistency with the other plot commands is corrected in Maple 9.

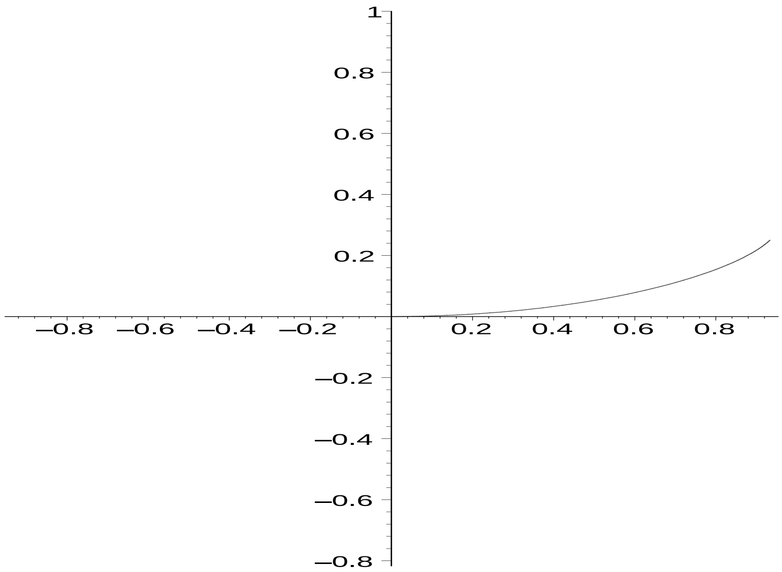
7.6 Creating Animations

A Maple animation is formed when the elements of a list of Maple plots is displayed in rapid succession. To animate the drawing of a function
$r = \cos(5\,\theta)$
for $0 < \theta < \pi$ in polar coordinates, create a sequence of plots of the function over shorter time intervals, say $0 < \theta < \theta_0$ for $\theta_0 = \frac{\pi}{12}, \frac{\pi}{6}, \frac{\pi}{4}, ..., \pi$.

```
> polarframes := seq( plot( sin(5*theta), theta = 0 ..  theta0,
                            coords=polar ),
                     theta0 = [ n * Pi/12 $ n = 1 ..  12 ] ):
```

The `display` command from the `plots` package will display the plots as a list with the `insequence=true` option. The animation is, of course, playable only in an active Maple worksheet (or a worksheet that has been exported to HTML). To play the animation, click the left mouse button anywhere in the first frame of the animation. Then use the VCR control buttons on the context bar to advance through the frames individually, once from beginning to end, or continuously.

```
> display( [ polarframes ], insequence=true );
```

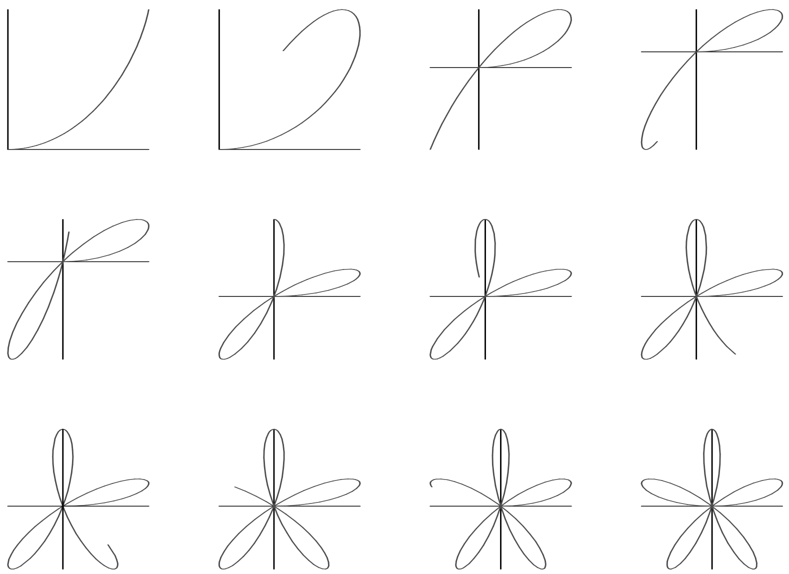


For a hardcopy of this animation it makes more sense to display all 12 frames of the animation as an array (or matrix) with three rows each containing four plots. The `matrix` command, from the `linalg` package, is used to create the 3 x 4 array of plots, then the `display` command is used to display the result. The optional argument `tickmarks=[0,0]` suppresses the tickmarks on both axes (they become too cluttered to be of any use).

```
> with( linalg ):

Warning, the protected names norm and trace have been redefined and
unprotected
```

```
> display( matrix( 3, 4, [polarframes]), tickmarks=[0,0] );
```



The concluding example in this section illustrates the relationship between the unit circle and the sine curve. The basic idea is to create separate plots of the sine curve and the unit circle centered at (-1,0). The animation comes with the addition of the line segment from the center of the circle to a point on the circle and the line segment from this point on the circle to the corresponding point on the sine curve for different angles between 0 and 2π. Do not be concerned if some of the details are unclear at first. Once you see the animation and look at the individual pieces used to compose the individual frames, the process will become much less mysterious.

The unit circle with center (-1,0) can be created with the `circle` command from the `plottools` package.

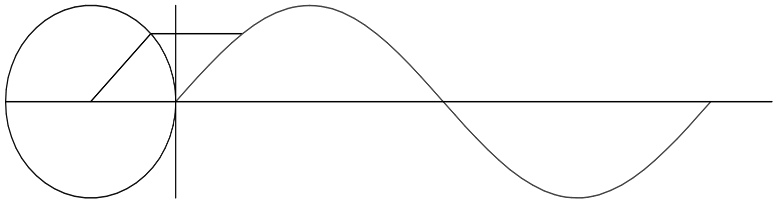
```
> with( plottools ):
Warning, the name arrow has been redefined
> p1 := circle( [-1,0], 1, color=blue ):
> p2 := plot( sin(x), x=0 ..  2*Pi, color=red ):
```

User-defined functions are used to create the line segments from the center of the circle, (-1,0), to a point on the circle, (cos(t)-1, sin(t)), to the point on the sine curve, (t, sin(t)), for an arbitrary angle and the composite plot showing the circle, sine curve, and line segments..

```
> circ_pt_line := t -> plot( [ [-1,0], [cos(t)-1,sin(t)], [t,sin(t)] ],
                           color=black ):
> composite_plot := t -> display( [ p1, p2, circ_pt_line(t) ],
                              view=[-2..7,-1..1], tickmarks=[0,0],
                              scaling=constrained ):
```

The frame of the movie with $t = \frac{\pi}{4}$ can now be created with the command

```
> composite_plot( Pi/4 );
```



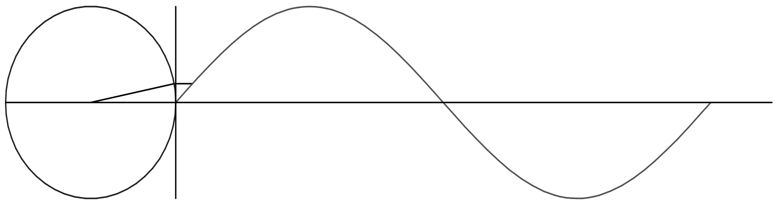
The list of 32 uniformly distributed angles between 0 and 2π is

```
> angles := [ i*Pi/16 $ i = 1 ..  32 ];
```

$$angles := [\frac{\pi}{16}, \frac{\pi}{8}, \frac{3\pi}{16}, \frac{\pi}{4}, \frac{5\pi}{16}, \frac{3\pi}{8}, \frac{7\pi}{16}, \frac{\pi}{2}, \frac{9\pi}{16}, \frac{5\pi}{8}, \frac{11\pi}{16}, \frac{3\pi}{4}, \frac{13\pi}{16}, \frac{7\pi}{8}, \frac{15\pi}{16}, \pi,$$
$$\frac{17\pi}{16}, \frac{9\pi}{8}, \frac{19\pi}{16}, \frac{5\pi}{4}, \frac{21\pi}{16}, \frac{11\pi}{8}, \frac{23\pi}{16}, \frac{3\pi}{2}, \frac{25\pi}{16}, \frac{13\pi}{8}, \frac{27\pi}{16}, \frac{7\pi}{4}, \frac{29\pi}{16},$$
$$\frac{15\pi}{8}, \frac{31\pi}{16}, 2\pi]$$

We conclude by making a frame for each of these angles and animating the resulting list of composite plots.

```
> movieframes := [ seq( composite_plot( theta ), theta=angles ) ]:
> display( movieframes, insequence=true, scaling=constrained );
```



7.7 Three Types of Brackets in Maple

Three types of brackets have been described in this worksheet: `( ... )`, `[ ... ]`, and `{ ... }`.

Parentheses or round brackets, `( ... )`, are used for the mathematical grouping of terms, including the specification of function arguments.

```
> 7 * (3+4);
```

$$49$$

```
> sin( Pi/4 );
```

$$\frac{\sqrt{2}}{2}$$

Square brackets, `[ ... ]`, are used for creating lists.

```
> [ seq( n^2, n=-2..2 ) ];
```

$$[4, 1, 0, 1, 4]$$

Curly brackets, { ... }, are used for creating sets.

```
> { seq( n^2, n=-2..2 ) };
```

$$\{0, 1, 4\}$$

Do not attempt to interchange these symbols! For example,

```
> sin[ Pi/4 ];
```

$$\sin_{\frac{\pi}{4}}$$

```
> exp({3});
```

```
Error, invalid input: exp expects its 1st argument, x, to be of type
algebraic, but received {3}
```

7.8 Common Problems and How to Fix Them

A Using a Command That Is Not Known by the Maple Kernel

Some commands are defined as part of a *package* that is not automatically loaded into the Maple kernel. If one of these commands is used prior to loading the package, the output will simply echo the input after the arguments have been simplified (if possible). For example,

```
> restart;
> completesquare( x^2 + 2*x + 2 );
```

$$\mathrm{completesquare}(x^2 + 2\,x + 2)$$

```
> with( student );
```

[D, *Diff*, *Doubleint*, *Int*, *Limit*, *Lineint*, *Product*, *Sum*, *Tripleint*, *changevar*, *completesquare*, *distance*, *equate*, *integrand*, *intercept*, *intparts*, *leftbox*, *leftsum*, *makeproc*, *middlebox*, *middlesum*, *midpoint*, *powsubs*, *rightbox*, *rightsum*, *showtangent*, *simpson*, *slope*, *summand*, *trapezoid*]

```
> completesquare( x^2 + 2*x + 2 );
```

$$(x + 1)^2 + 1$$

B Using Reserved Words and Protected Names

Maple places relatively few restrictions on the names that can be used for objects. When such a name is used, Maple generates an appropriate error message.

```
> union := {1,2,3};
```

```
Error, reserved words `union` or `minus` unexpected
```

```
> Pi := 22/7;
Error, attempting to assign to 'Pi' which is protected
```

Be forewarned, however, that if a value is assigned to a name that is also a Maple command, then the new assignment overwrites the command definition. This feature can be used — by knowledgeable and brave users — to extend or customize the functionality of some of Maple's built-in commands.

7.9 Maple and Calculus

The techniques introduced in the earlier parts of this section will now be applied to the fundamental calculus concepts: limit, derivative, and integral and their applications.

A Limits

```
> restart;
```

The Maple command to compute $\lim_{x\to a} \mathrm{f}(x)$ is `limit( f(x), x=a );`.

```
> limit( sin(x)/x, x=0 );
```

$$1$$

```
> limit( (1+x)^(1/x), x=0 );
```

$$e$$

To obtain a limit at ∞, use `infinity`.

```
> limit( (1+x/n)^n, n=infinity );
```

$$e^x$$

Right- or left-hand limits are returned when the (optional) third argument is `right` or `left`, respectively.

```
> limit( tan(x), x=Pi/2 );
```

$$\mathit{undefined}$$

```
> limit( tan(x), x=Pi/2, right );
```

$$-\infty$$

```
> limit( tan(x), x=Pi/2, left );
```

$$\infty$$

B Derivatives and Tangent Lines

```
> restart;
```

Given a function f, the slope of the secant lines through (x, f(x)) and ($x+h$, f($x+h$)) is the difference quotient. As h approaches 0, these quantities become the slope of the tangent line at (x, f(x)). The slope of the secant line can be obtained with the Maple command

```
> m[secant] := (f(x+h) - f(x) ) / h;
```

$$m_{secant} := \frac{\mathrm{f}(x+h) - \mathrm{f}(x)}{h}$$

In practice, it is often necessary to use the simplify command to force Maple to `simplify` the difference quotient. For example,

```
> f := x -> x^3 - 3*x^2 - 4;
```

$$f := x \to x^3 - 3\,x^2 - 4$$

```
> m[secant];
```

$$\frac{(x+h)^3 - 3\,(x+h)^2 - x^3 + 3\,x^2}{h}$$

```
> simplify( m[secant] );
```

$$3\,x^2 + 3\,x\,h + h^2 - 6\,x - 3\,h$$

```
> m[tangent] := limit( m[secant], h=0 );
```

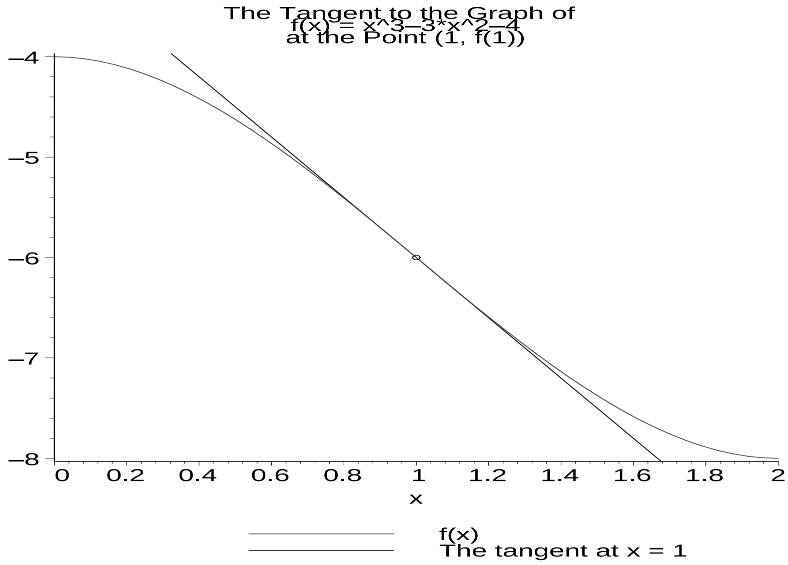
$$m_{tangent} := 3\,x^2 - 6\,x$$

The `Tangent` command from the `Student[Calculus1]` package provides a simple means to display a function and the tangent line at a point.

```
> with( Student[Calculus1] );
```

[*AntiderivativePlot*, *ApproximateInt*, *ArcLength*, *Asymptotes*, *Clear*, *CriticalPoints*, *DerivativePlot*, *ExtremePoints*, *FunctionAverage*, *FunctionChart*, *GetMessage*, *GetNumProblems*, *GetProblem*, *Hint*, *InflectionPoints*, *Integrand*, *InversePlot*, *MeanValueTheorem*, *NewtonQuotient*, *NewtonsMethod*, *PointInterpolation*, *RiemannSum*, *RollesTheorem*, *Roots*, *Rule*, *Show*, *ShowIncomplete*, *ShowSteps*, *Summand*, *SurfaceOfRevolution*, *Tangent*, *TaylorApproximation*, *Understand*, *Undo*, *VolumeOfRevolution*, *WhatProblem*]

```
> Tangent( f(x), x=1, output=plot );
```



The equation of this tangent line is obtained by changing the third argument to `output=line`.

```
> Tangent( f(x), x=1, output=line );
```

$$-3\,x - 3$$

The **TangentLinePlot** maplet [`MapletViewer`] [`MapleNet`] provides a customized user interface to the `Tangent` command. The `MapleNet` version of this maplet is available over the WWW to anyone using a computer with a reasonably current version of `Java`. The `MapletViewer` version requires that Maple 8 (or newer) is loaded on your local computer.

The derivative of an expression is obtained with the `diff` command.

```
> fp := diff( f(x), x );
```

$$fp := 3\,x^2 - 6\,x$$

The `D` command acts like the differentiation operator in that it computes the derivative of a function and returns a function.

```
> Df := D(f);
```

$$Df := x \to 3\,x^2 - 6\,x$$

Note that the latter form is more convenient when it will be necessary to compute the derivative at a specific point.

```
> Df(1) = eval( fp, x=1 );
```

$$-3 = -3$$

Higher-order derivatives are obtained by specifying additional arguments to the `diff` command or by composing `D` with itself an appropriate number of times.

```
> diff( exp(a*x), x, x );
```

$$a^2\, e^{(a\, x)}$$

```
> diff( exp(a*x), x$5 );
```

$$a^5\, e^{(a\, x)}$$

C Applications of Derivatives

Implicit Differentiation

Implicit differentiation can be done in several different ways. Consider the implicitly defined function

```
> impl_eq := (x-1)^4 = x^2 - y^2;
```

$$impl\_eq := (x-1)^4 = x^2 - y^2$$

The `implicitdiff` command can be used to find derivatives by implicit differentiation.

```
> dy/dx = implicitdiff( impl_eq, y, x );
```

$$\frac{dy}{dx} = -\frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{y}$$

Observe that the second and third arguments define the dependence between the names – the second argument is the dependent variable, the third argument is the independent variable.

The same result can be found by rewriting the implicit function to explicitly note the dependence of y on x

```
> eq2 := eval( impl_eq, y=y(x) );
```

$$eq2 := (x-1)^4 = x^2 - \mathrm{y}(x)^2$$

and differentiating the result with respect to x.

```
> q1 := diff( eq2, x );
```

$$q1 := 4\,(x-1)^3 = 2\,x - 2\,\mathrm{y}(x)\,(\tfrac{d}{dx}\,\mathrm{y}(x))$$

```
> q2 := solve( q1, { diff(y(x),x) } );
```

$$q2 := \{\tfrac{d}{dx}\,\mathrm{y}(x) = -\frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\mathrm{y}(x)}\}$$

These two commands can, of course, be combined into a single command.

```
> impl_diff := diff( eval( impl_eq, y=y(x) ), x );
```

$$impl\_diff := 4\,(x-1)^3 = 2\,x - 2\,\mathrm{y}(x)\,(\tfrac{d}{dx}\,\mathrm{y}(x))$$

The formula for $\frac{d}{dx}\,\mathrm{y}(x)$ is obtained by solving the above equation for this quantity.

```
> impl_dydx := isolate( impl_diff,  diff(y(x),x) );
```

$$impl\_dydx := \frac{d}{dx}\,\mathrm{y}(x) = \frac{1}{2}\,\frac{-4\,(x-1)^3 + 2\,x}{\mathrm{y}(x)}$$

While this formula is not the same as either of the derivatives found earlier, it is easily seen to be an equivalent result.

```
> simplify( impl_dydx );
```

$$\frac{d}{dx}\,\mathrm{y}(x) = -\frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\mathrm{y}(x)}$$

Higher-order implicit derivatives are obtained with `implicitdiff` by repeating the independent variable as many times as the desired order of the derivative (exactly as in `diff`).

```
> d2y/dx2 = implicitdiff( impl_eq, y, x,x );
```

$$\frac{d2y}{dx2} = -\frac{6\,y^2\,x^2 - 12\,y^2\,x + 5\,y^2 + 4\,x^6 - 24\,x^5 + 56\,x^4 - 68\,x^3 + 49\,x^2 - 20\,x + 4}{y^3}$$

Compare this with the step-by-step computation of the same (we hope) result with `diff` :

```
> q3 :=diff( eq2, x$2 );
```

$$q3 := 12\,(x-1)^2 = 2 - 2\,(\tfrac{d}{dx}\,\mathrm{y}(x))^2 - 2\,\mathrm{y}(x)\,(\tfrac{d^2}{dx^2}\,\mathrm{y}(x))$$

```
> q4 := solve( q3, { diff(y(x),x,x) } );
```

$$q4 := \left\{\frac{d^2}{dx^2}\,\mathrm{y}(x) = -\frac{6\,x^2 - 12\,x + 5 + (\frac{d}{dx}\,\mathrm{y}(x))^2}{\mathrm{y}(x)}\right\}$$

```
> lhs(q4[]) = simplify( eval( rhs(q4[]), q2 ) );
```

$$\frac{d^2}{dx^2}\,\mathrm{y}(x) = -\frac{6\,\mathrm{y}(x)^2\,x^2 - 12\,\mathrm{y}(x)^2\,x + 5\,\mathrm{y}(x)^2 + 4\,x^6 - 24\,x^5 + 56\,x^4 - 68\,x^3 + 49\,x^2 - 20\,x + 4}{\mathrm{y}(x)^3}$$

Linearization of a Function

```
> restart;
> with( Student[Calculus1] ):
```

The linearization of a function f(x) at $x = a$ is L(x) = f(a) + f'(a) ($x - a$). For example, if

```
> f := x -> sqrt( 1+x );
```

$$f := x \rightarrow \sqrt{1+x}$$

and

```
> a := 0;
```

$$a := 0$$

the linearization of f at $x = a$ can be obtained with the `TaylorApproximation` command from the `Student[Calculus1]` package as follows:

```
> TaylorApproximation( f(x), x=a );
```

$$1+\frac{x}{2}$$

The `unapply` command can be sued to convert this into a function

```
> L1 := unapply( TaylorApproximation( f(x), x=a ), x );
```

$$L1 := x \to 1+\frac{1}{2}\,x$$

An alternate definition of this linearization is

```
> L2 := x -> f(a) + D(f)(a)*(x-a);
```

$$L2 := x \to \mathrm{f}(a)+\mathrm{D}(f)(a)\,(x-a)$$

The difference in the output is caused by the fact that Maple does not fully evaluate the right-hand side of the arrow operator (->) until the function is actually called. What this means is that if either the function f or the point a are changed, then the function `L2` will reflect those changes as well. The function `L1`, however, uses the values of f and a <u>at the time the command is executed</u> and is unaffected by subsequent changes. This is illustrated in the following examples.

```
> L1(b) = L2(b);
```

$$1+\frac{b}{2}=1+\frac{b}{2}$$

```
> a := 1;
```

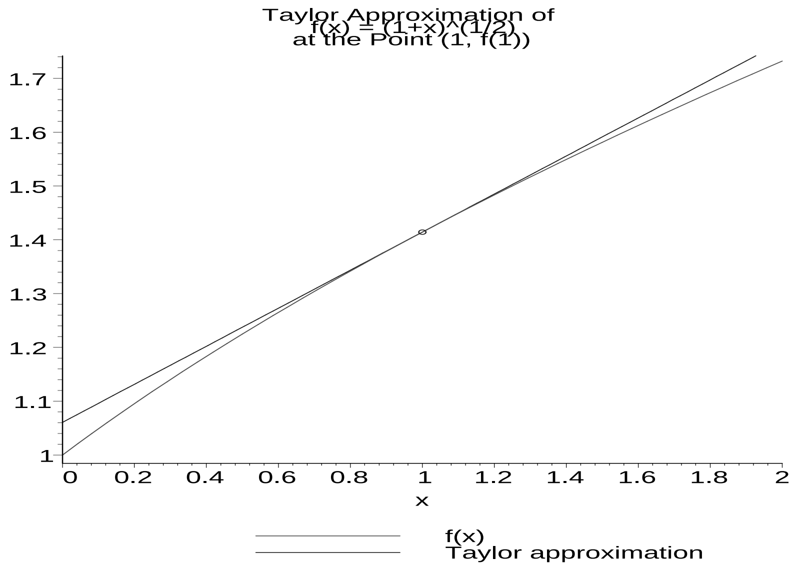
$$a := 1$$

```
> L1(b) <> L2(b);
```

$$1+\frac{b}{2}\neq\sqrt{2}+\frac{\sqrt{2}\,(b-1)}{4}$$

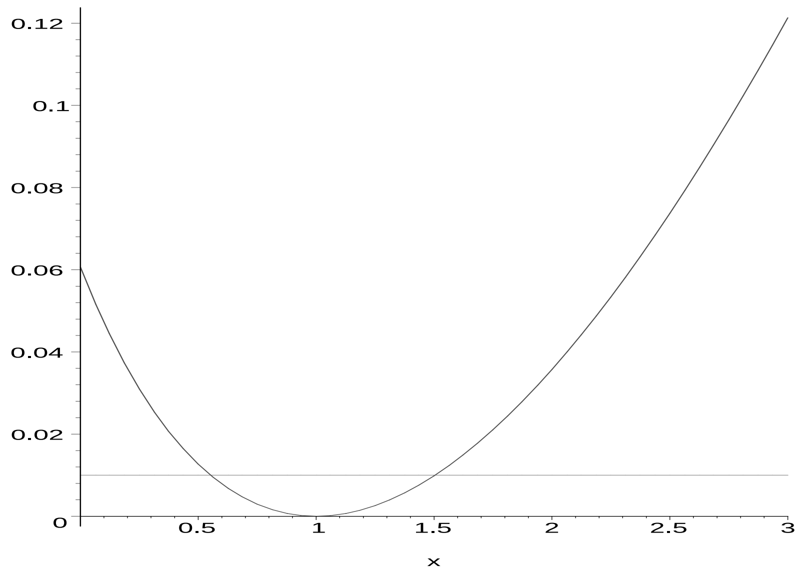
The function and its linearization can be graphed as follows.

```
> TaylorApproximation( f(x), x=a, output=plot );
```



To determine the largest interval on which the linearization differs from the original function by no more than a specified amount, *i.e.*, to estimate the largest δ such that $|x - a| < \delta$ implies $|\mathrm{f}(x) - \mathrm{L}(x)| < \varepsilon$, we first look at the graph

```
> epsilon := 0.01:
> plot( [ abs( f(x)-L2(x) ), epsilon ], x = 0 ..  3 );
```



Click the left mouse button as close as possible to the leftmost of the two intersections of the curves in the plot. The box at the left end of the context bar shows the approximate coordinates of this point; this gives the point (0.57, 0.01). Repeating this process for the rightmost intersection gives (1.49, 0.01). This gives the estimate $\delta = \min($ 1-0.57, 1.49-1 $) = 0.43$.

More accurate estimates of the intersection points will give more accurate estimates for δ. In this case Maple can solve the inequality $|\mathrm{f}(x) - \mathrm{L}(x)| < \varepsilon$.

```
> solve( abs( f(x)-L2(x) ) < epsilon, x );
```

$$\mathrm{RealRange}(\mathrm{Open}(0.5526014252), \mathrm{Open}(1.503967117))$$

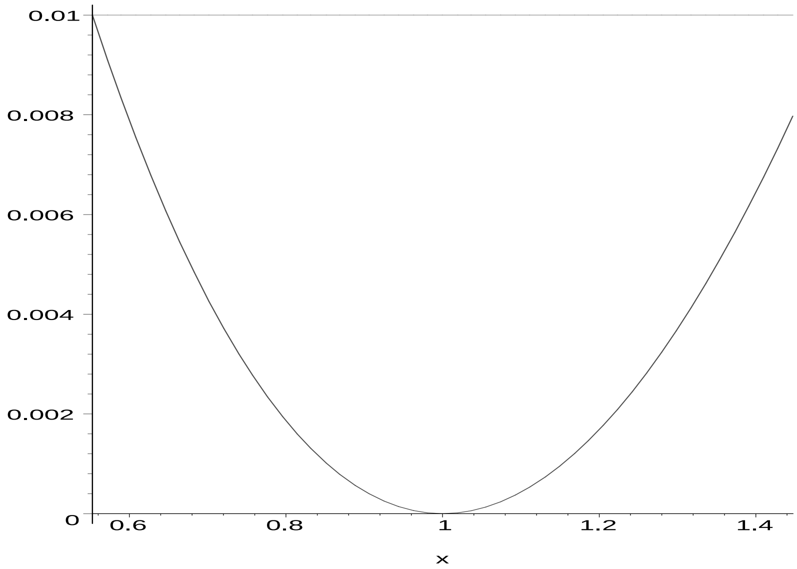
This yields

```
> delta := min( 1-0.5526014252, 1.503967117-1 );
```

$$\delta := 0.4473985748$$

To conclude, check that the error never exceeds ε over this interval.

```
> plot( [ abs( f(x)-L2(x) ), epsilon ], x = 1-delta ..  1+delta );
```



Newton's Method

```
> restart;
> with( Student[Calculus1] ):
```

Newton's Method is implemented with the `NewtonsMethod` command (from the `Student[Calculus1]` package). For example, consider the problem of finding the solution closest to $x = -1$, accurate to five decimal places, to $\mathrm{f}(x) = 0$ when

```
> f := x -> cos( 5*x ) - x:
```

Define the initial guess

```
> x0 := -1;
```

$$x0 := -1$$

Apply one iteration of Newton's Method to obtain

```
> x1 := NewtonsMethod( f(x), x=x0, iterations=1 );
```

$$x1 := -0.7784735010$$

Additional iterations of Newton's Method yield

```
> x2 := NewtonsMethod( f(x), x=x1, iterations=1 );
```

$$x2 := -0.7677474241$$

```
> x3 := NewtonsMethod( f(x), x=x2, iterations=1 );
```

$$x3 := -0.7674935683$$

```
> x4 := NewtonsMethod( f(x), x=x3, iterations=1 );
```

$$x4 := -0.7674934212$$

Thus, $x = -.76749$ is an approximate value of the root closest to $x = -1$. If this process had continued too much longer, it would be more efficient to use the following loop to compute the iterates. (See the online help for `do .. end do` .)

```
> x[0] := -1;
```

$$x_0 := -1$$

```
> x[1] := NewtonsMethod( f(x), x=x[0], iterations=1 );
```

$$x_1 := -0.7784735010$$

```
> for n from 2 while abs( x[n-1] - x[n-2] ) >= 10^(-6) do
>   x[n] := NewtonsMethod( f(x), x=x[n-1], iterations=1 );
> end do;
```

$$x_2 := -0.7677474241$$
$$x_3 := -0.7674935683$$
$$x_4 := -0.7674934212$$

The explicit function used for one iteration of Newton's Method can be defined with

```
> g := unapply( NewtonsMethod( f(x), x=X, iterations=1 ), X );
```

$$g := X \to X - \frac{1.\,(\cos(5.\,X) - 1.\,X)}{-5.\sin(5.\,X) - 1.}$$

A sequence of Newton iterates can be obtained by adding `output=sequence` :

```
> NewtonsMethod( f(x), x=-1, iterations=3, output=sequence );
```

$$-1, -0.7784735010, -0.7677474241, -0.7674935683$$

If the `iterations` argument is omitted, then 5 iterations are performed.

```
> NewtonsMethod( f(x), x=-1, output=sequence );
```

$$-1, -0.7784735010, -0.7677474241, -0.7674935683, -0.7674934212, -0.7674934213$$

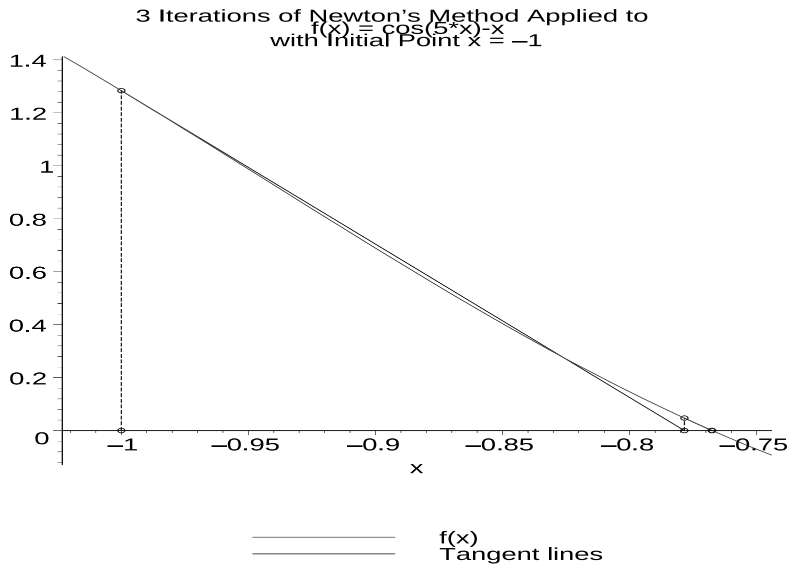
If the `output=sequence` argument is also omitted, then only the fifth iteration is returned

```
> NewtonsMethod( f(x), x=-1 );
```

$$-0.7674934213$$

To conclude this discussion, using `output=plot` produces in a graphical display of the Newton iterations.

```
> NewtonsMethod( f(x), x=-1, output=plot, iterations=3 );
```



Notice how each step of the Newton iteration uses the tangent line to the curve at the current iteration to obtain the next iteration. (To understand the hypotheses for the convergence theorem for Newton's method, think about what would happen if one of the tangent lines was horizontal, or nearly horizontal.)

D Definite, Indefinite, and Improper Integrals

```
> restart;
```

The `int` command is used to compute integrals in Maple. The indefinite integral $\int \ln(x)\,dx$ is obtained with the command

```
> int( ln(x), x ) + C;
```

$$x\ln(x) - x + C$$

Note that it is necessary to explicitly include the constant of integration, C.

If additional information is given, it should be possible to determine a specific value for C. For example, Maple can be used to find the function that satisfies the initial value problem

$$\text{y'(x)} = 5\,e^{(-3\,x)}$$

$\text{y}(0) = -10$

First, find the antiderivative (don't forget to include the constant of integration)

```
> eq1 := y = int( 5*exp(-3*x), x ) + C;
```

$$\mathit{eq1} := y = -\frac{5}{3}\,e^{(-3\,x)} + C$$

To determine the appropriate value of C, apply the initial condition

```
> eq2 := eval( eq1, { y=-10, x=0 } );
```

$$\mathit{eq2} := -10 = -\frac{5}{3} + C$$

and solve for C

```
> eq3 := C = solve( eq2, C );
```

$$\mathit{eq3} := C = \frac{-25}{3}$$

The solution to the initial value problem is

```
> eval( eq1, eq3 );
```

$$y = -\frac{5}{3}\,e^{(-3\,x)} - \frac{25}{3}$$

The `Student[Calculus1]` package contains a number of commands for the visualization and estimation of Riemann sums.

```
>  with( Student[Calculus1] ):
```

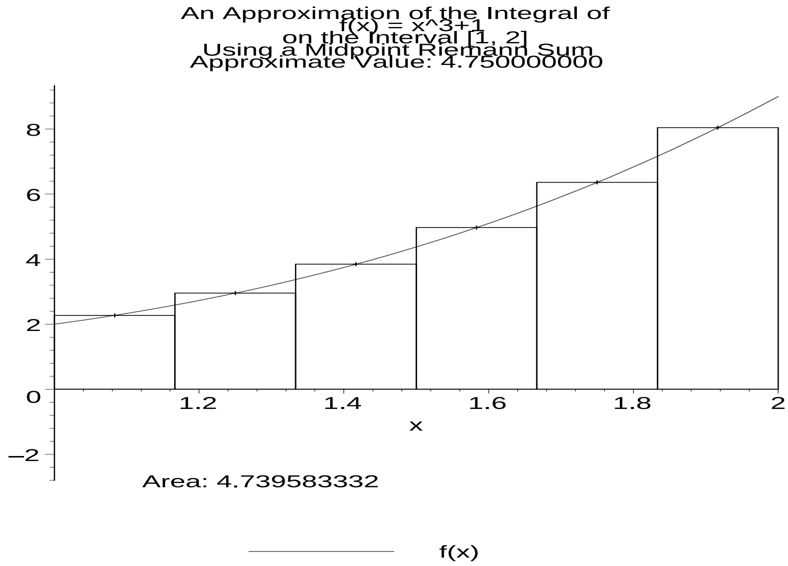
To illustrate, consider the problem of approximating the value of $\int_1^2 x^3 + 1\,dx$. Let

```
>  f := x^3 + 1;
```

$$f := x^3 + 1$$

The midpoint approximation to this integral with 6 subdivisions is the area of the rectangles displayed with the `ApproximateInt` command

```
>  ApproximateInt(f, x=1..2, method=midpoint, partition=6, output=plot);
```



Observe that this plot displays the value of the corresponding Riemann sum (4.73958332) and the value of the definite integral (4.75). The explicit Riemann sum for this example can be obtained with

```
>  ApproximateInt(f, x=1..2, method=midpoint, partition=6, output=sum);
```

$$\frac{1}{6}\left(\sum_{i=0}^{5}\left(\left(\frac{13}{12}+\frac{i}{6}\right)^3+1\right)\right)$$

The `value` of this Riemann sum can be obtained as either

```
>  value( % );
```

$$\frac{455}{96}$$

or

```
>  ApproximateInt( f, x=1..2, method=midpoint, partition=6 );
```

$$\frac{455}{96}$$

The floating-point approximation to this value is

```
> evalf( % );
```

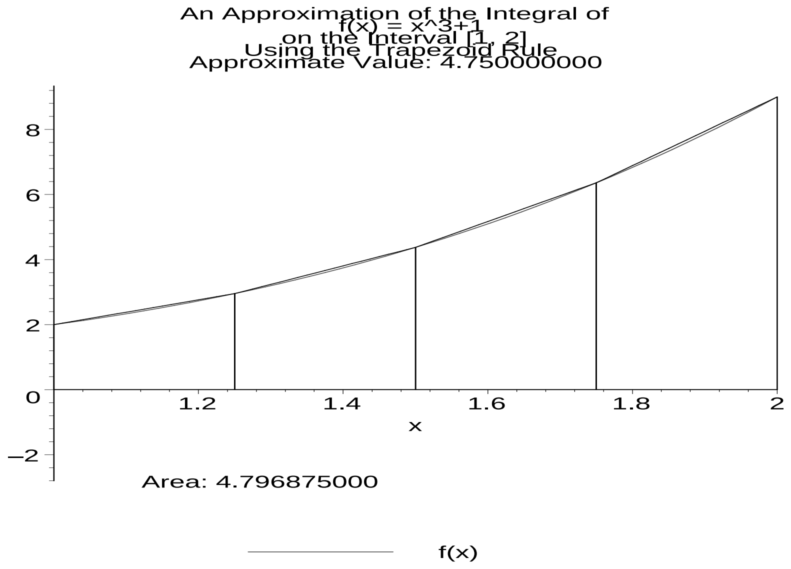
$$4.739583333$$

An animation showing the convergence of a Riemann sum, or other quadrature method, can be obtained as follows:

```
> ApproximateInt( f, x=1..2, output=animation,
                  method=random, subpartition=all, refinement=random,
                  partition=2, iterations=8 ):
```

For other Riemann sums, change the `method` option to `right`, `left`, `random`, `upper`, `lower`, or `random`. For the Trapezoidal and Simpson's Rules, use `method=trapezoid` or `method=simpson`, respectively.

```
> ApproximateInt(f, x=1..2, method=trapezoid, partition=4, output=plot);
```



The exact value of the definite integral is

```
> int( f, x = 1 .. 2 );
```

$$\frac{19}{4}$$

```
> evalf( % );
```

$$4.750000000$$

While Maple is designed to be able to automatically evaluate many integrals, it can also be used to assist with techniques including substitution and integration by parts. The explicit commands implementing this capability are quite complex. But, they are easy to use in the form of the **Differentiation** maplet [`MapletViewer`] [`MapleNet`].

E Applications of Integrals

Arclength of a Smooth Curve

```
> restart;
> with( Student[Calculus1] ):
```

The definite integral for the arclength of a smooth curve is usually easy to formulate but difficult to evaluate. The general formula for the arclength of the graph of $y = \mathrm{f}(x)$ for x in $[a, b]$ is

```
> ArcLength( f(x), x=a .. b, output=integral );
```

$$\int_a^b \sqrt{(\frac{d}{dx}\,\mathrm{f}(x))^2 + 1}\, dx$$

For example, with

```
> f := x -> 2*x^(3/2);
```

$$f := x \to 2\,x^{(3/2)}$$

on the interval [0, 1], the definite integral for the arclength is

```
> ArcLength( f(x), x=0..1, output=integral );
```

$$\int_0^1 \sqrt{9\,x + 1}\, dx$$

The exact value of this integral is

```
> ArcLength( f(x), x=0..1 );
```

$$\frac{20\sqrt{10}}{27} - \frac{2}{27}$$

Maple is able to evaluate many more integrals than most humans, but there are many arclength integrals that cannot be evaluated explicitly. If this occurs, or if Maple's answer involves functions that are not familiar to you, use `evalf` to force a numerical approximation of the integral.

```
> f := x -> sin(x);
```

$$f := \sin$$

```
> ArcLength( f(x), x=0..1 );
```

$$\sqrt{2}\,\mathrm{EllipticE}(\sqrt{1 - \cos(1)^2},\, \frac{\sqrt{2}}{2})$$

```
> evalf( % );
```

$$1.311442498$$

Improper Integrals

```
> restart;
```

Improper integrals are not a problem for Maple. Here are three examples.

```
> int( x * exp(-x), x = 1 .. infinity );
```

$$2\,e^{(-1)}$$

```
> int( 1/sqrt(4-x^2), x = 0 .. 2 );
```

$$\frac{\pi}{2}$$

```
> int( 1/x, x = 1 .. infinity );
```

$$\infty$$

Multiple Integrals

```
> restart;
```

The `int` and `Int` commands can be used to construct iterated integrals corresponding to double and triple integrals.

```
> Int( Int( exp(x-y), x = y .. 1 ), y = 0 .. 1 );
```

$$\int_0^1 \int_y^1 e^{(x-y)}\, dx\, dy$$

```
> % = value( % );
```

$$\int_0^1 \int_y^1 e^{(x-y)}\, dx\, dy = -2 + e$$

The area of the ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ can be found by parameterizing the ellipse in polar coordinates: $x = a\, r \cos(\theta)$, $y = b\, r \sin(\theta)$ with $0 <= r <= 1$ and $0 <= \theta < 2\pi$. The Jacobian of this change of variables is $a\, b\, r$, so the area of the ellipse is

```
> Int( Int( a*b*r, r = 0 .. 1 ), theta = 0 .. 2*Pi );
```

$$\int_0^{2\pi} \int_0^1 a\, b\, r\, dr\, d\theta$$

```
> % = value( % );
```

$$\int_0^{2\pi} \int_0^1 a\, b\, r\, dr\, d\theta = a\, b\, \pi$$

Note that this result reduces to the familiar area of a circle when $a = b$.

7.10 Maple and Differential Equations

```
> restart;
```

The `dsolve` command attempts to solve a differential equation. If initial conditions are provided, a particular solution is sought; otherwise, the successful result will be the general solution.

```
> ode := diff( y(x), x ) = y(x) * (3-y(x)) * (x-1);
```

$$ode := \frac{d}{dx}\, \mathrm{y}(x) = \mathrm{y}(x)\, (3 - \mathrm{y}(x))\, (x - 1)$$

```
> soln := dsolve( ode, y(x) );
```

$$soln := \mathrm{y}(x) = \frac{3}{1 + 3\, e^{(-3/2\, x^2 + 3\, x)}\, \_C1}$$

Note that Maple has returned the general solution to this differential equation. The name `_C1` is the parameter for this one-dimensional family of solutions.

To find the solution that passes through a specific point, substitute the initial condition into the solution, solve for `_C1`, and substitute the result back into the general solution:

```
> q1 := eval( soln, { y(x) = 2, x=0 } );
```

$$q1 := 2 = \frac{3}{1 + 3\_C1}$$

```
> q2 := solve( q1, { _C1 } );
```

$$q2 := \{\_C1 = \frac{1}{6}\}$$

```
> q3 := eval( soln,  q2 );
```

$$q3 := \mathrm{y}(x) = \frac{3}{1 + \frac{1}{2}\, e^{(-3/2\, x^2 + 3\, x)}}$$

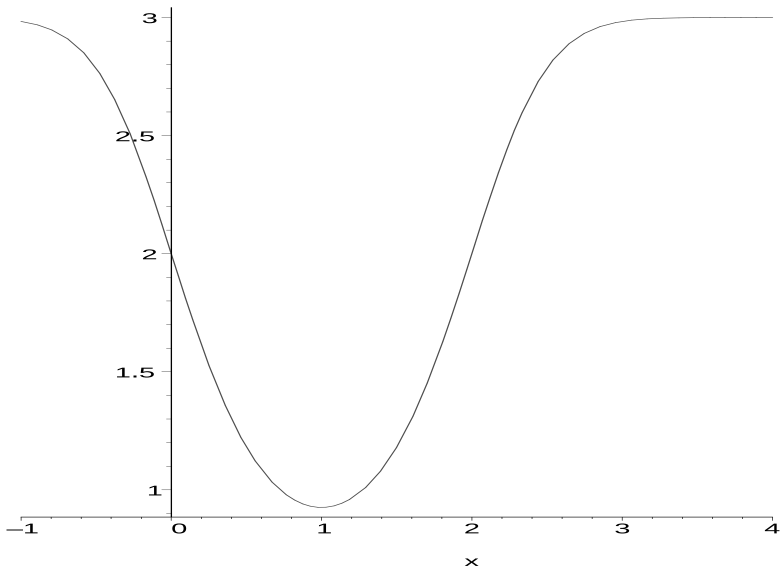
If the particular solution to an initial value problem is all that is needed, then the above process can be streamlined to a single `dsolve` command in which the first argument is a *set* containing the differential equation and the initial condition.

```
> q4 := dsolve( { ode, y(0)=2 }, y(x) );
```

$$q4 := \mathrm{y}(x) = \frac{3}{1 + \frac{1}{2}\, e^{(-3/2\, x^2 + 3\, x)}}$$

Notice that the solution returned by `dsolve` is an equation. To plot this solution, use the `rhs` command to pass only the right-hand side of the solution to the `plot` command.

```
> plot( rhs(q4), x=-1..4 );
```



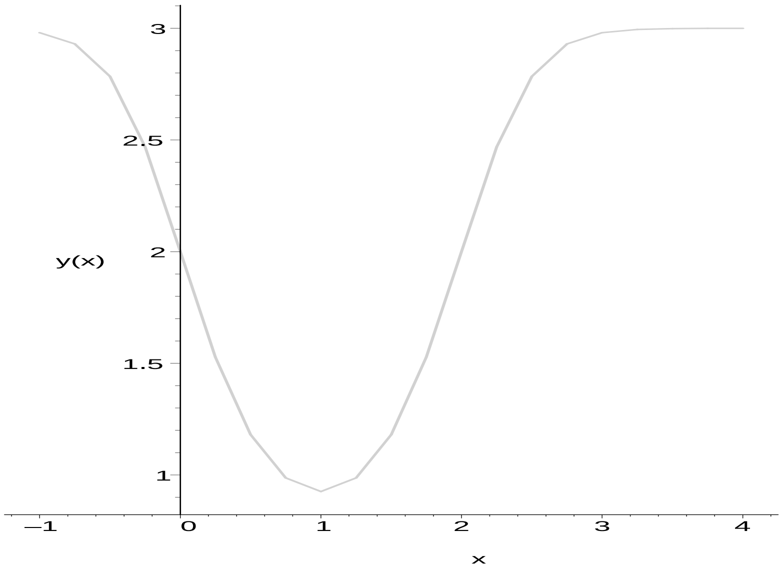
The DEtools package contains a number of useful commands for working with differential equations.

```
> with( DEtools );
```

[DEnormal, DEplot, DEplot3d, DEplot_polygon, DFactor, DFactorLCLM, DFactorsols, Dchangevar, GCRD, LCLM, MeijerGsols, PDEchangecoords, RiemannPsols, Xchange, Xcommutator, Xgauge, abelsol, adjoint, autonomous, bernoullisol, buildsol, buildsym, canoni, caseplot, casesplit, checkrank, chinisol, clairautsol, constcoeffsols, convertAlg, convertsys, dalembertsol, dcoeffs, de2diffop, dfieldplot, diffop2de, dpolyform, dsubs, eigenring, endomorphism_charpoly, equinv, eta_k, eulersols, exactsol, expsols, exterior_power, firint, firtest, formal_sol, gen_exp, generate_ic, genhomosol, gensys, hamilton_eqs, hypergeomsols, hyperode, indicialeq, infgen, initialdata, integrate_sols, intfactor, invariants, kovacicsols, leftdivision, liesol, line_int, linearsol, matrixDE, matrix_riccati, maxdimsystems, moser_reduce, muchange, mult, mutest, newton_polygon, normalG2, odeadvisor, odepde, parametricsol, phaseportrait, poincare, polysols, power_equivalent, ratsols, redode, reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rifread, rifsimp, rightdivision, rtaylor, separablesol, solve_group, super_reduce, symgen, symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom]

In particular, the DEplot command can be used to display a direction field and/or solution curves for a differential equation.

```
> DEplot( ode, [y(x)], x=-1..4, [ [0,2] ], arrows=none );
```

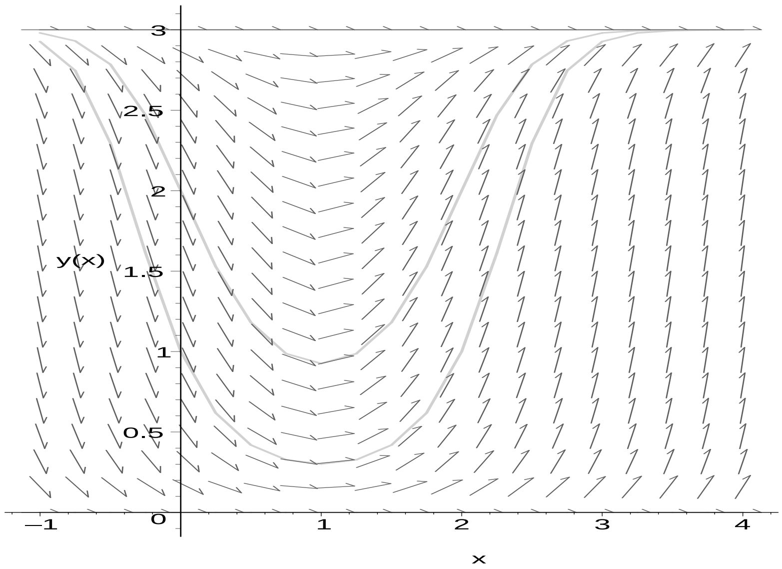


Changing the last argument to arrows=thin will include the direction field. Adding additional ordered pairs to the list of initial conditions will add additional solution curves to the plot. The online help for DEplot describes other options that control features such as the color of the arrows and solution curves, the window for the plot, and the stepsize.

```
> init_cond := [ [0,0], [0,1], [0,2], [0,3] ];
```

$$init\_cond := [[0, 0], [0, 1], [0, 2], [0, 3]]$$

```
> DEplot( ode, [y(x)], x=-1..4, init_cond, arrows=thin );
```



Chapter 8

Tips for Maple Users

This portion of the manual contains tips and suggestions that will help you, and your students, make optimal use of Maple while learning Calculus.[1] These suggestions begin with a few "rules" that will aid new users in the process of developing good Maple usage habits. A worksheet containing the examples presented in Tables 8.1 – 8.8 can be found at the WWW at

`http://www.math.sc.edu/~meade/141L-F03/manual/tip.mws` .

[1]The tips in this chapter are based on a similar list prepared in collaboration with Robert Lopez for *Applied Engineering Mathematics*, (Addison Wesley Longman, 2002).

Tip #1 *Avoid potential name clashes.*
Never assign to a variable on the left if you are using that variable as a "free variable" on the right. This avoids name clashes later. (See Table 8.1.)

Tip #2 *Avoid assigning values to common single-letter names.*
One suggestion is to use names such as `q`, `q1`, `q2`, ... because `q` is one of the more infrequently used mathematical names and it is easy to find on the keyboard. Adherence to this tip is particularly important if the name is frequently used as a "working variable" on the right-hand side of other expressions. (See Table 8.2.)

Tip #3 *Never reassign something (involving itself) to itself.*
If you do, you will have two input regions containing the same name assigned two different values, and you may not know which version of the name you want (or will receive) during later usage. (See Table 8.3.)

Tip #4 *Never use Maple's ditto operator to reference a previous output.*
Use of the ditto operators, `%`, `%%`, and `%%%`, can generate an unreadable worksheet because there is no indication to subsequent readers of the order in which commands are executed. (It is still possible to create an incomprehensible worksheet without `%`, but `%` is one bad actor that we should exclude from our plays.) If you really want to have a shorthand notation for previous results, use Maple's `history` command. Use `?history` to see the corresponding help document. (See Table 8.4.)

Tip #5 *Avoid making assumptions unless it is absolutely necessary.*
The problem with assumptions is that they have lasting, and on-going, side effects. When an assumption is used, try to revert to an unassumed name as soon as it becomes feasible. The `assuming` feature provides a more useful and effective way to make temporary assumptions about variables. (See Table 8.5.)

Tip #6 *Avoid using the `op` command or the selection operator, `[ ]`, to refer to elements of a set.*
Because the elements of a set are unordered, it is likely that the set will have a different appearance if the commands are re-executed. Recommended methods to access elements of a set generally use one or more of the commands `eval`, `select`, `remove`, `selectremove`, and `subs` commands. Another alternative is to convert the set into a list and use `sort` to put the elements of the list in a predetermined order. (See Table 8.6.)

Tip #7 *Use a unique index variable in a `do .. end do` command.*
When the `do` command terminates, the index variable retains its final value. A simple method for creating a unique index variable is to change the case or prepend (or append) one or more characters to the name, *e.g.*, `X`, or `IND_x` instead of `x`. If the index variable is used outside the `do` command, it should be restored to an unevaluated name with the `unassign` command.

The `seq`, `add`, `mul`, `sum`, and `product` commands also use an index variable. However, the index variable used by these commands is local to the command. This means the value of a global variable with the same name as the index variable is unchanged by these commands. This approach is preferred because these commands are generally more efficient than the corresponding `do .. end do` command. (See Table 8.7.)

Tip #8 *Start all Maple sessions with the `restart` command.*
In addition to putting a `restart` command at the beginning of every worksheet, it is recommended to load all packages at the top of the worksheet. (See Table 8.8.)

<table>
<tr><th>Improper Usage</th><th>Recommended Usage</th></tr>
<tr><td>> restart;
> fn := x -> exp(x^2);
$fn := x \to e^{(x^2)}$
> x := a+3;
$x := a+3$
> diff(fn(x), x);
Error, wrong number (or type) of
parameters in function diff
> fn(x), x;
$e^{((a+3)^2)}, a+3$</td><td>> restart;
> fn := x -> exp(x^2);
$fn := x \to e^{(x^2)}$
> Dfn := diff(fn(x), x);
$Dfn := 2\,x\,e^{(x^2)}$
> eval(Dfn, x=a+3);
$2\,(a+3)\,e^{((a+3)^2)}$
> fn(x), x;
$e^{(x^2)}, x$</td></tr>
<tr><td>> restart;
> x := t;
y := t^2;
$x := t$
$y := t^2$
> solve({ x=x, y=y }, { y, t });
$\{t = t, t^2 = t^2\}$</td><td>> restart;
> EQx := x = t;
EQy := y = t^2;
$EQx := x = t$
$EQy := y = t^2$
> solve({ EQx, EQy }, { y, t });
$\{t = x, y = x^2\}$</td></tr>
</table>

Table 8.1: Two illustrations of the problems that can arise when names clash.

<table>
<tr><th>Improper Usage</th><th>Recommended Usage</th></tr>
<tr><td>> restart;
> f := 1;
$f := 1$
> for n from 1 to 2 do
f := simplify(int(f*exp(x),x));
end do;
$f := e^x$
$f := \frac{1}{2}\,e^{(2\,x)}$
> for n from 1 to 2 do
f := simplify(int(f*exp(x),x));
end do;
$f := \frac{1}{6}\,e^{(3\,x)}$
$f := \frac{1}{24}\,e^{(4\,x)}$</td><td>> restart;
> f0 := 1;
$f0 := 1$
> for n from 1 to 2 do
f||n := simplify(
int(exp(x)*f||(n-1),x));
end do;
$f1 := e^x$
$f2 := \frac{1}{2}\,e^{(2\,x)}$
> for n from 1 to 2 do
f||n := simplify(
int(exp(x)*f||(n-1),x));
end do;
$f1 := e^x$
$f2 := \frac{1}{2}\,e^{(2\,x)}$</td></tr>
</table>

Table 8.2: Illustration of the problems that can arise when "working variables" are used as Maple names.

<table>
<tr><th>Improper Usage</th><th>Recommended Usage</th></tr>
<tr><td>
> <code>restart;</code>
> <code>fn := (u,v) -> (u+v)^2 - u*v;</code>
$fn := (u,\, v) \to (u+v)^2 - u\, v$
> <code>u := 2;</code>
$u := 2$
> <code>fn(u,v);</code>
$(2+v)^2 - 2\, v$
> <code>solve(fn(u,v)=0, {u});</code>
<code>Error, (in solve) a constant</code>
<code>is invalid as a variable, 2</code>
</td><td>
> <code>restart;</code>
> <code>fn := (u,v) -> (u+v)^2 - u*v;</code>
$fn := (u,\, v) \to (u+v)^2 - u\, v$
> <code>u0 := 2;</code>
$u0 := 2$
> <code>solve(fn(u,v)=0, {u});</code>
$\{u = (-\frac{1}{2} + \frac{1}{2}\, I\, \sqrt{3})\, v\},\ \{u = (-\frac{1}{2} - \frac{1}{2}\, I\, \sqrt{3})\, v\}$
</td></tr>
</table>

Table 8.3: Illustration of the problems that can arise when a name is used on both sides of an assignment. Note the different results when the `do .. end do` commands are executed the second time in the left column.

<table>
<tr><th>Improper Usage</th><th>Recommended Usage</th></tr>
<tr><td>
> <code>restart;</code>
> <code>f := 1;</code>
$f := 1$
> <code>for n from 1 to 2 do</code>
<code> f := simplify(int(exp(x)*%,x));</code>
<code>end do;</code>
$f := e^x$
$f := \frac{1}{2}\, e^{(2\, x)}$
> <code>for n from 1 to 2 do</code>
<code> f := simplify(int(exp(x)*%,x));</code>
<code>end do;</code>
$f := \frac{1}{6}\, e^{(3\, x)}$
$f := \frac{1}{24}\, e^{(4\, x)}$
</td><td>
> <code>restart;</code>
> <code>f0 := 1;</code>
$f0 := 1$
> <code>for n from 1 to 2 do</code>
<code> f||n := simplify(</code>
<code>   int(exp(x)*f||(n-1),x));</code>
<code>end do;</code>
$f1 := e^x$
$f2 := \frac{1}{2}\, e^{(2\, x)}$
> <code>for n from 1 to 2 do</code>
<code> f||n := simplify(</code>
<code>   int(exp(x)*f||(n-1),x));</code>
<code>end do;</code>
$f1 := e^x$
$f2 := \frac{1}{2}\, e^{(2\, x)}$
</td></tr>
</table>

Table 8.4: Illustration of the problems that can arise from the use of the ditto operator, `%`.

Improper Usage	Recommended Usage

`> restart;`
`> int( exp(a*x), x=0..infinity );`
`Definite integration: Can't determine`
`if the integral is convergent.`
`Need to know the sign of --> -a`
`Will now try indefinite integration`
`and then take limits.`
$\lim_{x\to\infty} \frac{e^{(a\,x)}-1}{a}$
`> assume( a < 0 );`
`> int( exp(a*x), x=0..infinity );`
$-\frac{1}{a^{\sim}}$
`> cutoff := piecewise( a>0, 1 );`
$\mathit{cutoff} := \mathrm{PIECEWISE}()$
	`> restart;`
	`> int( exp(a*x), x=0..infinity );`
	`Definite integration: Can't determine`
	`if the integral is convergent.`
	`Need to know the sign of --> -a`
	`Will now try indefinite integration`
	`and then take limits.`
	$\lim_{x\to\infty} \frac{e^{(a\,x)}-1}{a}$
	`> assume( a < 0 );`
	`> int( exp(a*x), x=0..infinity );`
	$-\frac{1}{a^{\sim}}$
	`> unassign( 'a' );`
	`> cutoff := piecewise( a>0, 1 );`
	$\mathit{cutoff} := \begin{cases} 1 & 0 < a \\ 0 & \mathit{otherwise} \end{cases}$

`> restart;`
`> q:=Int( x*sin(n*x), x=-Pi..Pi);`
$q := \int_{-\pi}^{\pi} x \sin(n\,x)\,dx$
`> assume( n, integer );`
`> a:=value( q );`
$a := -\frac{2\,\pi\,(-1)^{n^{\sim}}}{n^{\sim}}$
`> add( a*sin(n*x), n=1..2 );`
$-\frac{2\,\pi\,(-1)^{n^{\sim}}\sin(x)}{n^{\sim}} - \frac{2\,\pi\,(-1)^{n^{\sim}}\sin(2\,x)}{n^{\sim}}$
	`> restart;`
	`> q:=Int( x*sin(n*x), x=-Pi..Pi);`
	$q := \int_{-\pi}^{\pi} x \sin(n\,x)\,dx$
	`> a:=simplify( value(q) )`
	`assuming n::integer;`
	$a := \frac{2\,(-1)^{(1+n)}\,\pi}{n}$
	`> add( a*sin(n*x), n=1..3 );`
	$2\,\pi\sin(x) - \pi\sin(2\,x) + \frac{2}{3}\,\pi\sin(3\,x)$

Table 8.5: Two illustrations of the problems when assumptions are used inappropriately.

Improper Usage	Recommended Usage

`> restart;`
`> eqn := x^4 = 16;`
$eqn := x^4 = 16$
`> soln := { solve( eqn, x ) };`
$soln := \{-2, 2, 2\,I, -2\,I\}$
`> xRE := soln[1], soln[2];`
$xRE := -2, 2$
`> xREp := xRE[2];`
$xREp := 2$
`> xREn := xRE[1];`
$xREn := -2$
	`> restart;`
	`> eqn := x^4 = 16;`
	$eqn := x^4 = 16$
	`> soln := { solve( eqn, x ) };`
	$soln := \{-2, 2, 2\,I, -2\,I\}$
	`> xRE := remove( type, soln,`
	`nonreal );`
	$xRE := \{-2, 2\}$
	`> xREp := select( type, xRE,`
	`positive );`
	$xREp := \{2\}$

`> restart;`
`> ode := diff(x(t),t)=-y(t),`
`diff(y(t),t)= x(t);`
$ode := \frac{d}{dt}\,\mathrm{x}(t) = -\mathrm{y}(t),\ \frac{d}{dt}\,\mathrm{y}(t) = \mathrm{x}(t)$
`> ic := x(0) = 1, y(0) = 0;`
$ic := \mathrm{x}(0) = 1, \mathrm{y}(0) = 0$
`> S := dsolve( { ode, ic },`
`{ x(t), y(t) } );`
$S := \{\mathrm{y}(t) = \sin(t), \mathrm{x}(t) = \cos(t)\}$
`> X := rhs( S[1] );`
$X := \sin(t)$
`> Y := rhs( S[2] );`
$Y := \cos(t)$
	`> restart;`
	`> ode := diff(x(t),t)=-y(t),`
	`diff(y(t),t)= x(t);`
	$ode := \frac{d}{dt}\,\mathrm{x}(t) = -\mathrm{y}(t),\ \frac{d}{dt}\,\mathrm{y}(t) = \mathrm{x}(t)$
	`> ic := x(0) = 1, y(0) = 0;`
	$ic := \mathrm{x}(0) = 1, \mathrm{y}(0) = 0$
	`> S := dsolve( { ode, ic },`
	`{ x(t), y(t) } );`
	$S := \{\mathrm{x}(t) = \cos(t), \mathrm{y}(t) = \sin(t)\}$
	`> X := eval( x(t), S );`
	$X := \cos(t)$
	`> Y := eval( y(t), S );`
	$Y := \sin(t)$

Table 8.6: Two examples where the assumption that the elements of a *set* are ordered can lead to problems. The results from the two different approaches are *sometimes* consistent, the assignments to `xRE`, `xREp`, `xREn`, `X`, and `Y` in the left column all *assume* a specific ordering of the results in `soln` and `S`.

Improper Usage	Recommended Usage

`> restart;`
`> f := x -> exp(x/2):`
`> a := 0:  b := 2:  h := 0.01:`
`> Rsum := 0.0:`
`for x from a to b by h do`
`  Rsum := Rsum + h*f(x)`
`end do:`
`Rsum;`
3.455162226
`> diff( f(x), x );`
`Error, wrong number (or type) of parameters in function diff`
`> x;`
2.01
	`> restart;`
	`> f := x -> exp(x/2):`
	`> a := 0:  b := 2:  h := 0.01:`
	`> Rsum := 0.0:`
	`for x from a to b by h do`
	`  Rsum := Rsum + h*f(x)`
	`end do:`
	`Rsum;`
	3.455162226
	`> unassign( 'x' );`
	`> x;`
	x
	`> diff( f(x), x );`
	$\frac{1}{2}\,e^{(\frac{x}{2})}$
	`> Rsum := add( h*f(x), x=[seq(h*i, i=0..200)] );`
	$Rsum := 3.455162226$
	`> x,i;`
	x, i
	`> i := 0:`
	`> Rsum := add( h*f(h*i), i=0..200 );`
	$Rsum := 3.455162226$
	`> i;`
	0

Table 8.7: Example of the problems that can occur when a common variable name is used as the index variable in a `do .. end do` construction. The first recommended solution illustrates how to remove the assignment from the index counter. The second and third solutions demonstrate how the computation can be done with commands that use a local index.

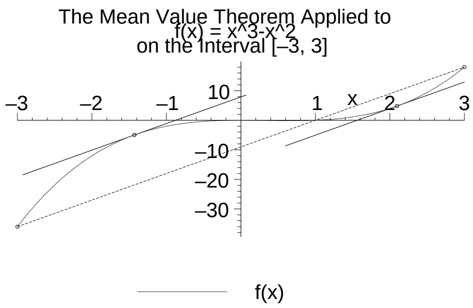
Improper Usage	Recommended Usage
> `f := x^3 - x^2;` $f := x^3 - x^2$ > `MeanValueTheorem( f, x=-3..3, output=plot );` $\text{MeanValueTheorem}(x^3 - x^2, x = -3..3, \mathit{output} = \mathit{plot})$ > `q1 := MeanValueTheorem( f, x=-3..3, output=points);` $\mathit{q1} := \text{MeanValueTheorem}(x^3 - x^2, x = -3..3, \mathit{output} = \mathit{points})$ > `evalf( q1 );` $\text{MeanValueTheorem}(x^3 - x^2, x = -3..3, \mathit{output} = \mathit{points})$	> `restart;` > `with( Student[Calculus1] ):` > `f := x^3-x^2;` $f := x^3 - x^2$ > `MeanValueTheorem( f, x=-3..3, output=plot );`  > `q1 := MeanValueTheorem( f, x=-3..3, output=points );` $\mathit{q1} := [\frac{1}{3} - \frac{2\sqrt{7}}{3}, \frac{1}{3} + \frac{2\sqrt{7}}{3}]$ > `evalf( q1 );` $[-1.430500874, 2.097167540]$

Table 8.8: An illustration of a typical problem encountered when the user forgets to load a package in a worksheet. One of the best ways to avoid this problem is to load all packages at the beginning of a worksheet, immediately following the `restart` command.

Chapter 9

Quick Reference Guide for Maple 8