Contents

1 Information for Instructors

Many of the exercise sets in Johnston and Mathews' *Calculus* contain one or more technology exercises marked with a special "T" icon. These exercises are intended to be solved with the use of a graphing calculator or computer algebra system (CAS). In addition, each chapter contains at least one Student Project. The text's Web site, `http://www.aw.com/johnston/`, contains Maple worksheets, *Mathematica* notebooks, and MATLAB M-files for these projects. See the Features section on page `vi` of the text's Preface for more details.

If you are considering having students use Maple for some of the Technology Exercises or Student Projects, please spend a few minutes reading this section. It contains general advice and suggestions for the successful integration of Maple into your course.

A Getting Started

A.1 Determine the Computing Needs of Your Students

First, you need to determine *where* your students will complete their Maple assignments. If students will be using on-campus labs, you will need to be sure that Maple is readily available in the lab, or labs, where students will be expected to work.

If possible, the labs should have the latest version of Maple; presently, Maple 7. Up-to-date information about updates, platforms, and hardware requirements can be found on the Waterloo Maple webpages at `http://www.maplesoft.com/products/Maple7/minsys.shtml`. If you do not yet have Maple on your campus, the name of your local sales representative can be obtained from the Waterloo Maple homepage at `http://www.maplesoft.com/sales/`.

If Maple is not available on campus, you will need to make arrangements for students to purchase individual copies of Maple. Actually, they should purchase the Student Edition of Maple. Additional information can be obtained by contaacting your Waterloo Maple sales representative or visiting the URL `http://www.maplesoft.com/products/Student/student.shtml`. You might want to check if campus bookstore(s) will stock the software for you. (Depending on local policies, making the software a "required course material" may allow students to use scholarship funds to help offset the purchase price.)

A.2 Familiarize Yourself with the Student Computing Environment

You should spend some time working with Maple in the same environment as your students. If they will be working in an on-campus lab, spend some time in the lab. Learn any special access procedures, how to launch Maple, check that you can access materials in a course archive or on the WWW, save a worksheet, print a worksheet, exit Maple, transfer worksheets via e-mail or FTP, and log out of the computer. Make a note of the hours the lab is open. Communicate this information to your students.

When, if ever, are lab assistants available? Do they have any knowledge of Maple? If not, you might want to provide them with some of the basics — including a copy of this document and the *Introduction to Maple* tutorial worksheet.

A.3 Install Maple Worksheets

Although students can access the Maple worksheets for the Student Projects on the Web site, it is recommended that these files be available directly from the on-campus labs. This allows students to access these materials even if Internet access is temporarily interrupted.

B Planning Your Course

B.1 Student Background

It is very helpful to collect some academic information about your students and to determine the amount and level of their prior experience with computers and Maple. Knowledge of their academic background and interests can be used to select appropriate Student Projects and to select more meaningful Technology Exercises for the class. If you plan to have students work in groups of two or more, you may want to group students with extensive experience with students with limited experience.

B.2 Integrating Computer Exercises into Your Syllabus

If assigning Technology Exercises and Student Projeccts is a new experience for you, it is recommended that you make the computer work a relatively small portion of the overall grade. Be very selective of the exercises and projects you assign and be sensitive to the amount of time it will take students to complete the computer-based assignments.

B.3 Familiarize Yourself with the Exercises and Projects

You are strongly encouraged to read Section 3 of this manual and to complete the tutorial worksheet, *An Introduction to Maple*, particularly if you are a new Maple user. Spend some time working through some of the Technology Exercises and Student Projects before the semester begins. This will provide a better feel for the projects and exercises that are most appropriate for your course and your students.

C Assigning Student Projects and Technology Exercises

C.1 Do the Exercises Before You Assign Them

Prior to assigning a Technology Exercise or a Student Project, solve the problem or complete the project on your own. Completing this work will enable you to determine if any additional instructions or hints need to be communicated to the students. Another, possibly more important, reason for completing the assignment in advance is to be sure you are prepared to answer any questions that the students might have about the assignment. This will also provide an opportunity for you to evaluate the length and level of difficulty of the assignment. Some exercises and projects that appear to be very straightforward require significantly more time and effort than you would expect. (The opposite is also true: some exercises that appear rather complicated and involved can be completed with relatively little time or effort.)

C.2 Discuss Maple Assignments in Class

When you are ready to assign a Technology Exercise or Student Project, you may want to provide the class with a brief overview of the assigned work. Discuss the relevance of the assignment to the topics currently or previously covered in the course. Neglecting to discuss computer assignments during class may give your students the ill-conceived idea that computer assignments are just "busy work" with no real importance to the lecture material, written homework assignments, or exams.

C.3 Assign Only Portions of Certain Projects

It is not necessary to assign an entire Student Project. You might find it appropriate to assign only portions of some of the projects. There may also be times when you decide to delete parts of a project in an effort to prevent the computer work from consuming an inappropriately large amount of the students' study time. The key is to customize each project to fit the particular needs of your course and your students.

C.4 Determine if Group Work is Allowed

Determine whether you will require, permit, or prohibit your students to work on the computer assignments in groups of more than one. As mentioned previously, one advantage of group work is that experienced computer users can help students with less experience adjust to the computer environment. In many instances Maple work benefits from group activities because the students who are not typing are able to catch syntax errors before the command is submitted to Maple. Of course, there is always the chance that one or two students will do all of the work for the entire group. A good way to avoid this problem is to include one or two short questions on each exam that relate to the computer assignments. Regardless of your decision for your course, be sure you clearly explain to the students whether you want them to work together or individually in the computer lab.

C.5 Determine Acceptable Forms for Completed Assignments

Clearly explain when and how computer assignments are to be submitted to you for grading. You probably do not want each student (or group) to turn in a hardcopy of all of their work relating to the project. Instead, have them prepare a formal project report that summarizes their work and answers the relevant questions.

Another possibility is to have students submit their work on a diskette or as an e-mail attachment. If you decide to have students submit work electronically, be sure to provide explicit instructions about the filename to use. For example, a combination of the student's last name, the chapter number, and the project letter would make a good filename, *e.g.*, `jones_ch3A`.

C.6 The *Information for Students* Section is Also for Instructors

Although the information in the following section is written for students, instructors 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 this section. However, if you are a new Maple user or have not used Maple 7, you should read this section more closely, directing particular attention to the terminology and descriptions of manipulations with Maple's graphical user interface.

2 Information for Students

The Student Projects and the Introduction to Maple tutorial worksheet can be found on the Web site as Maple *worksheets*. These worksheets, or "`.mws` files", are ASCII files that can be used with Maple on any platform — Windows, Macintosh, Linux, or Unix. Information is entered in a worksheet via the keyboard and Maple's graphical user worksheet. The purpose of this section is to introduce some of the essential facts and features of a Maple worksheet.

A Maple Worksheets

A.1 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 and controlled 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* (see Section A.8).

A *text region* contains information that is not executed by Maple. Typically this includes explanations and mathematical derivations related to the commands in 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.

Each execution group can contain any number of input and text regions, but no more than one *output region*. 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. (See Figure 2.)

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.

A.2 Executing Maple Commands

To execute the commands in an execution group use the left mouse button 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 either **Edit** ... **Execute** ... **Selection** or **Edit** ... **Execute** ... **Worksheet**. All output will be placed in an output region at the end of the execution group from which it was generated.

A.3 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 the Maple kernel 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.

A.4 Creating New Execution Groups

A new execution group is automatically 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 *tool bar*, located directly under the *menu bar*. 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 tool bar. (See Figure 1.)

Other conversions between input and text regions can be performed using the first four options in the **Insert** menu.

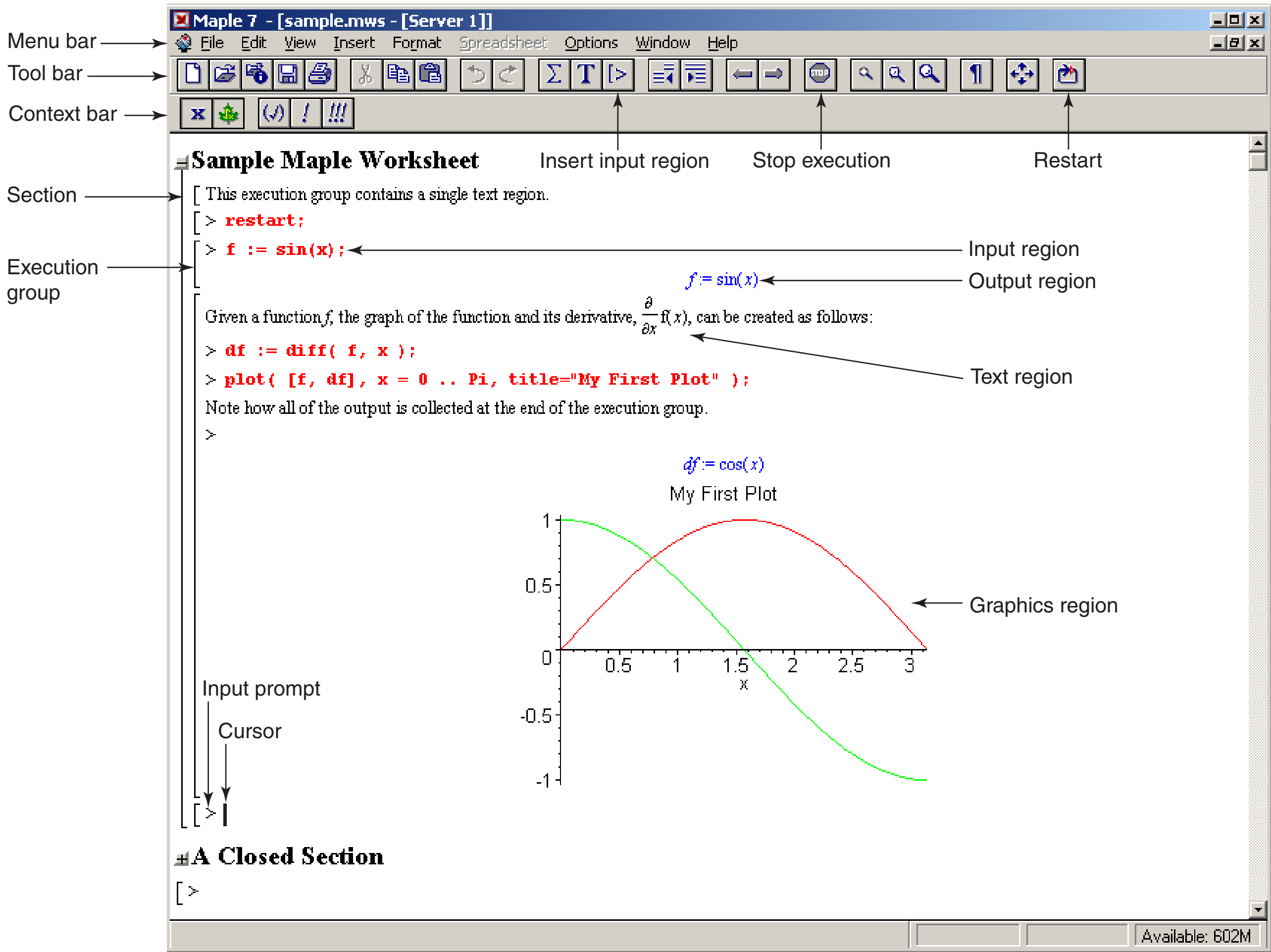


Figure 1: The different types of regions and sections in a Maple worksheet.

A.5 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 section 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. (See Figure 1.)

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. The tool bar contains cut, copy, and paste icons that can be used to duplicate or relocate a highlighted selection. Note that a section must be expanded in order for its contents to be successfully copied. Also, the contents of a Maple worksheet can be pasted into other applications, *e.g.*, Microsoft Word or PowerPoint.

A.6 Deleting Execution Groups and Sections

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

A.7 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`.

The bold **x** at the left end of the context bar toggles the current input region between Standard Math notation and Maple notation. *Standard Math notation* pretty-prints the command using as much mathematical notation as possible; *Maple notation* displays the formal Maple commands.

A.8 Context-Sensitive Menus

A *context-sensitive menu* is created whenever all or a portion of an output region is selected and the right mouse button is pressed (and held down). The items in the context-sensitive menu are operations that Maple determines to be the most pertinent common operations to perform on the selection. When one of these items is selected, the appropriate Maple 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.

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 original graph is re-executed.

A.9 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, *i.e.*, `alpha` for α, `exp(1)` for e, `infinity` for ∞, and `Pi` for π. The *matrix palette* can be used to create a template for a Maple `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 `%?`. Similarly, the *vector palette* can be used to create row or column vectors with up to four components. Note that the buttons on the matrix and vector palettes create a Maple `Matrix` and `Vector`, respectively, for use in the `LinearAlgebra` package, not a `matrix` and `vector` as used in the `linalg` package. The *expression palette* contains templates for common mathematical expressions ranging from sums, products, and quotients, to roots, exponentials, and trigonometric functions to limits, derivatives, and integrals (both definite and indefinite).

A.10 Saving Worksheets

A Maple worksheet can be saved by either selecting **File** ... **Save** or **File** ... **Save As ...** or clicking on the diskette icon in the tool bar. Be sure to use unique and descriptive filenames. It is also recommended that you save all Maple worksheets for this class in a single folder.

Worksheets should be saved frequently to minimize the amount of work lost in the event of an unexpected termination of Maple. Since it can be difficult to remember to do this, Maple 7 contains a feature that automatically saves a Maple worksheet at regular intervals (the default is every 3 minutes). This feature is activated via **Options** ... **AutoSave ....** Note that the auto-saved worksheet uses the original filename with the string "`_MAS`" appended.

A.11 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 document you require, you can use the `help` command, or the shorter name `?`, *e.g.*, `help(plot);` and `?plot` open the same Maple help document.

The help documents are built from sections and text, input, and output regions. Almost every help document contains an **Examples** section that shows the syntax, input, and output for the relevant command. While Maple help documents are very similar to Maple worksheets, 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.

The **Introduction** and **New User's Tour** selections on the **Help** menu provide links to additional introductory tutorials on a variety of topics including basic Maple terminology, the Maple worksheet interface, calculus, differential equations, linear algebra, numeric computations, graphics, and Maple programming.

B Frequently Encountered Problems

B.1 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. While this can be automated with the auto-save feature on the **Options** menu, it is a good habit to manually save a worksheet prior to printing.

B.2 Syntax Errors

Maple is a programming language that provides communication 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 2.)

Maple is case sensitive. The Maple constant **Pi** is different from the lower- and upper-case Greek letters **pi** and **PI**. (Only **Pi** is the mathematical constant π.)

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 execution group. You can type the remainder of the command in the new input region or reposition the cursor and correct or complete the original input region. When you next press `Return` in an execution group, all of the input regions in the 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.

B.3 Input Unchanged/Echoed

If the command `feval( sin(1) );` is executed Maple responds by echoing the command. This occurs because Maple does not recognize `feval` as a defined function. 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 the current Maple session with the `with` command.

B.4 No Output

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

If a command ends with a semi-colon and still does not produce any visible output, then the command has returned `NULL`, or the empty set. This might mean that an equation has no solution or just that Maple was unable to find a solution. The online help for the command generally explain the cases that produce a null response.

B.5 Printing Problems

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 approach to customize the printed output from Maple is to cut-and-paste the desired content to a separate Maple worksheet (or another application).

Bad page breaks are often found in worksheets containing plots or large matrices. This is because plots and matrices 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. The only way to see all page breaks without printing the worksheet is to use **File** . . . **Print Preview . . .** .

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

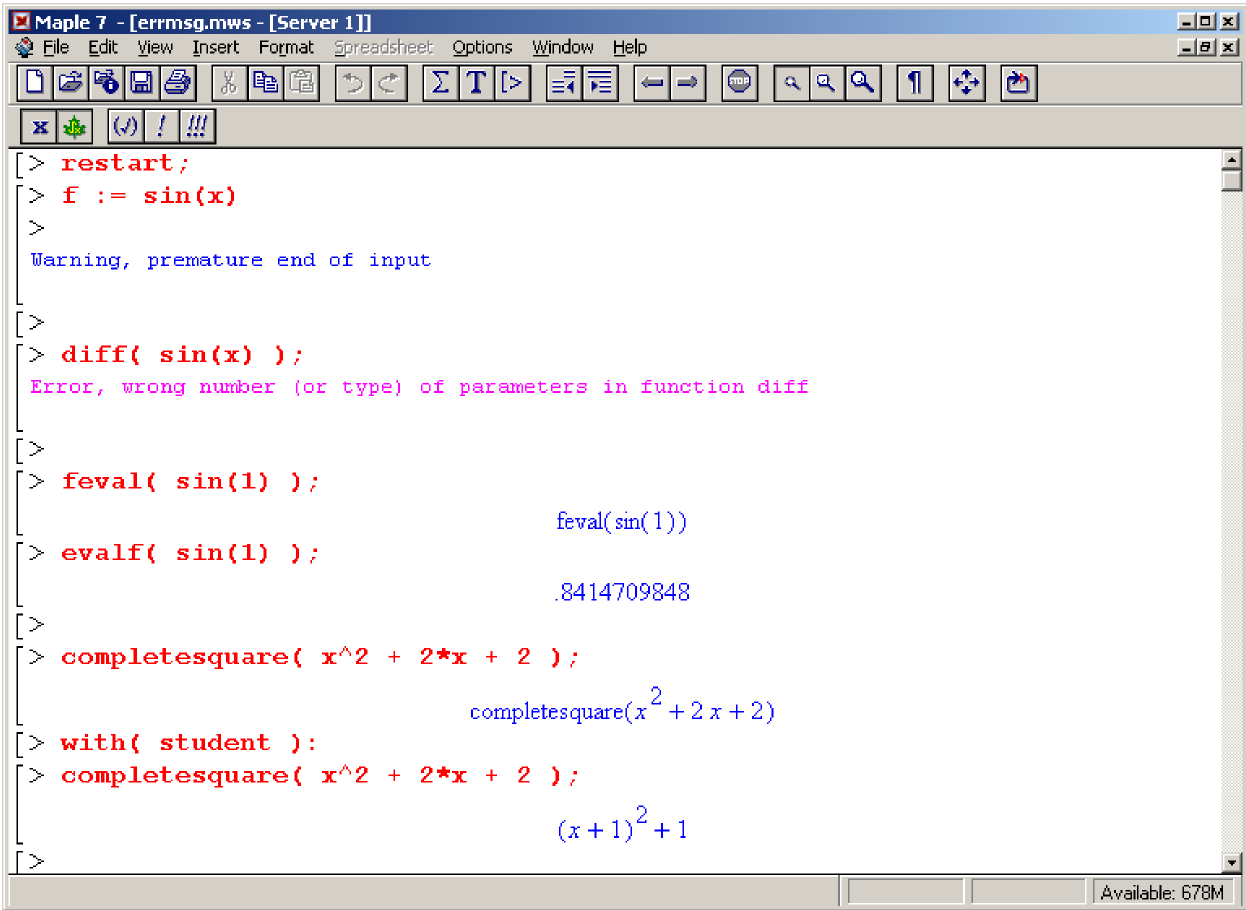


Figure 2: Examples of Maple warnings and error messages.

C Technology Exercises, Student Projects, and Maple

Many of the exercise sets in Johnston and Mathews' *Calculus* contain Technology Exercises. These exercises, identified by the "T" icon, are intended to be solved with the assistance of a graphing calculator or a CAS package such as Maple or *Mathematica*. A Maple worksheet is provided for each of the 21 Student Projects in the the text.

No student (or instructor) should consider solving (or assigning) one of these without an adequate background in Maple. The last section of this manual, "An Introduction to Maple", is intended to provide such a tutorial. In fact, the Maple worksheet from which this section was created is available on the Web site for the textbook, `http://www.aw.com/johnston/`.

3 An Introduction to Maple

A Introduction

Prior to attempting to use Maple, the novice user is encouraged to spend a few minutes reading through this introduction to become more familiar with some of the basic features, commands and structure of Maple. The Introduction to Maple worksheet that accompanies Johnston and Mathews' *Calculus* is an electronic version of this tutorial. This worksheet, named intro.mws, can be found on the CD that accompanies the text or at the text's Web site http://www.aw.com/johnston/.

B 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) arithmetic.

B.1 Addition, Subtraction, Multiplication, and Division

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

```
> 575754575849849885 + 7489498549859447495989 84;
```

$$748950430740520599448869$$

```
> 87575750 - 4897475988744894574949;
```

$$-4897475988744806999199$$

```
> 6868868686 * 18234987271740;
```

$$125253733060463458733640$$

```
> 99686868612732546598686500000000000000 / 5000;
```

$$19937373722546509319737300000000000$$

Remember that each command has to end with a semi-colon (or colon, if you do not wish to see the result).

B.2 Powers

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

```
> 55757 ^ 22;
```

$$2621722782213073468606173269872464991004411425248561190057363503\backslash$$
$$1073771467377455720035345978636911261049$$

```
> 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;
```

$$\infty$$

B.3 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**. Drag the palette to a position where it does not interfere with the current worksheet. (You may also need to resize the Maple and worksheet windows.)

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

$$151511$$

B.4 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 desired. For example, the three cube roots of 2 can be found exactly as

```
> soln := solve( x^3 = 2, x );
```

$$soln := 2^{(1/3)},\ -\frac{1}{2}\,2^{(1/3)} + \frac{1}{2}\,I\,\sqrt{3}\,2^{(1/3)},\ -\frac{1}{2}\,2^{(1/3)} - \frac{1}{2}\,I\,\sqrt{3}\,2^{(1/3)}$$

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).

```
> solve( x^3 = 2., x );
```

$$-.6299605250 - 1.091123636\,I,\ -.6299605250 + 1.091123636\,I,\ 1.259921050$$

The **evalf** command can also be used to force Maple to **eval**uate using **f**loating point arithmetic. Here the exact roots are evaluated using five digit floating point computations.

```
> evalf( soln );
```

$$1.259921050,\ -.6299605250 + 1.091123636\,I,\ -.6299605250 - 1.091123636\,I$$

By default, Maple performs all floating point computations using 10 significant digits. Including an index to the **evalf** command instructs Maple to use floating point numbers with the specified number of significant digits. (See also the online help for **Digits**.)

```
> evalf[5]( soln );
```

$$1.2599,\ -.62995 + 1.0912\,I,\ -.62995 - 1.0912\,I$$

The **fsolve** command can also be used to obtain floating point approximations to solutions of one or more equations. The basic syntax returns at most one real-valued solution.

```
> fsolve( x^3 = 2, x );
```

$$1.259921050$$

Additional arguments can be used to specify ranges to search (or avoid) for additional solutions. Or, for polynomial equations, the optional argument **complex** can be added to request approximations to all solutions.

```
> fsolve( x^3 = 2, x, complex );
```

$$-.6299605249 - 1.091123636\, I,\ -.6299605249 + 1.091123636\, I,\ 1.259921050$$

B.5 Using Previous Results

The percent symbol % 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, 4.806245936, can recalled with %%.

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

$$16.10000000$$

C Assigning Names

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 the 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; y := 3;
```

$$x := 2$$
$$y := 3$$

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 numeric 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 that have been made. To erase all assignments, it is easier to execute the **restart** command. The "restart" icon on the tool bar in Maple 7 makes this possible without inserting an execution group or any typing. (See Figure 1.)

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

$$x, y, prod$$

Note that if **prod** is defined as before, but prior to assigning values to **x** and **y**,

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

$$prod := x\,y$$

Now, 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 definition is made. In general, each name in an expression is "fully evaluated" prior to the execution of the assignment. If the resulting assignment contains one or more unevaluated names, all subsequent references to the assigned name use the current values of these names (if any) to determine the value of the assigned name.

C.1 Suppressing Output

The last few examples have had more than one command in each input region. These commands can appear on separate lines or on the same line. If you wish to suppress the display of results of a command, use a colon to terminate the command.

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

$$2700$$

D Maple Commands

```
> restart;
```

D.1 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 expression sequence 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 an expression sequence of numbers, then **min(** *exprseq* **)** returns 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 ];
```

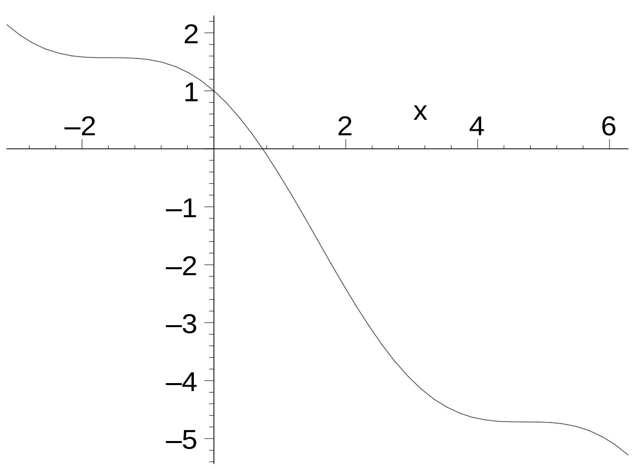
$$L := [2, 3, 6]$$

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

$$2$$

The following **plot** command will plot the function $\mathrm{f}(x) = \cos(x) - x$ on the interval $[\,-\pi, 2\,\pi]$.

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



In Maple, the equation $\cos(x) - x = 0$ is represented exactly as we would write 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 );
```

$$.7390851332$$

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

```
> evalf( Pi );
> evalf[50]( exp(1) );
```

$$3.141592654$$

$$2.7182818284590452353602874713526624977572470937000$$

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 );
```

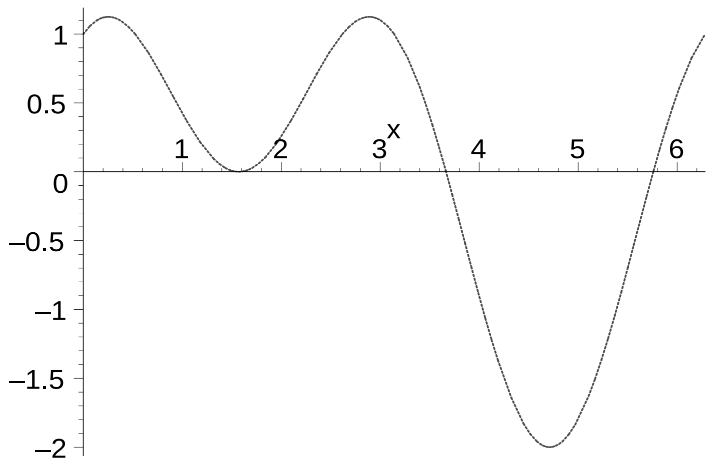
$$2\,I$$

Each of these quantities could also have been assembled using the palettes and context-sensitive menus as discussed previously in this worksheet.

D.2 Command Options

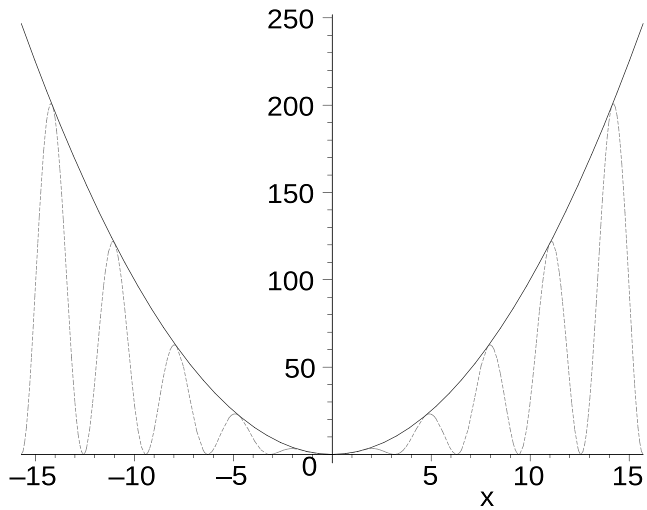
Many Maple commands, particularly the plotting commands, accept optional arguments for customizing the output. For example, the options **thickness=3** and **linestyle=7** plots the command using a thicker dotted line (see the online help for **plot[options]**).

```
> plot( sin(x) + cos(2*x), x = 0..2*Pi, thickness=3, linestyle = 7 );
```



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 ] );
```



D.3 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. 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 above methods are applicable only when a reasonable guess of the Maple command name is known. In other situations the **Help** menu can be used to browse the online help database.

D.4 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. One of the more common packages is the **plots** package.

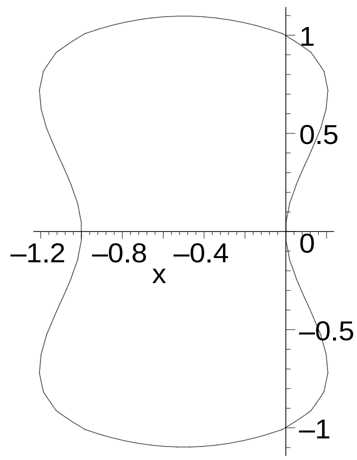
```
> with( plots );
Warning, the name changecoords has been redefined
```

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

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.

D.5 Defining Functions

You can also create your own Maple commands, including Maple implementations of mathematical functions. For example, the function $f(x) = x^2 \sin(x)^2$ 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 );
```

$$fred^2 \sin(fred)^2$$

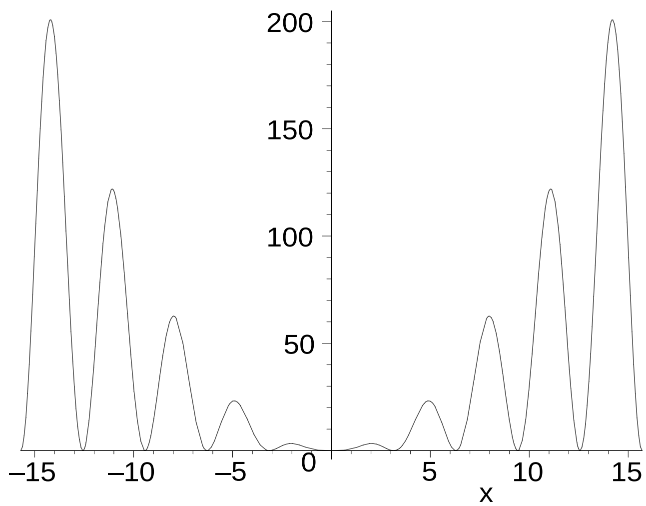
```
>  f( Pi/2 );
```

$$\frac{1}{4}\pi^2$$

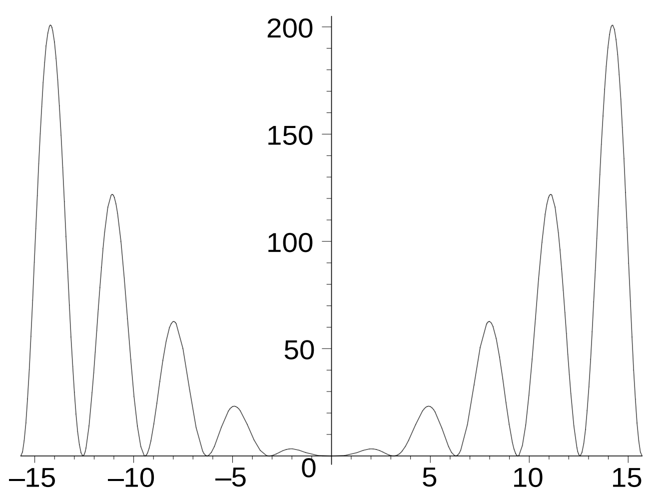
Notice how Maple automatically simplifies the value of the function if possible.

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.

It should be noted that the first form is not appropriate when the function definition contains a conditional that cannot be evaluated for a symbolic argument. For example, consider the piecewise-defined function

```
> F := x -> if x<0 then -x else sqrt(x) end if:
```

When this function is evaluated with a symbolic argument, an error is returned because Maple is unable to decide which branch of the conditional to follow.

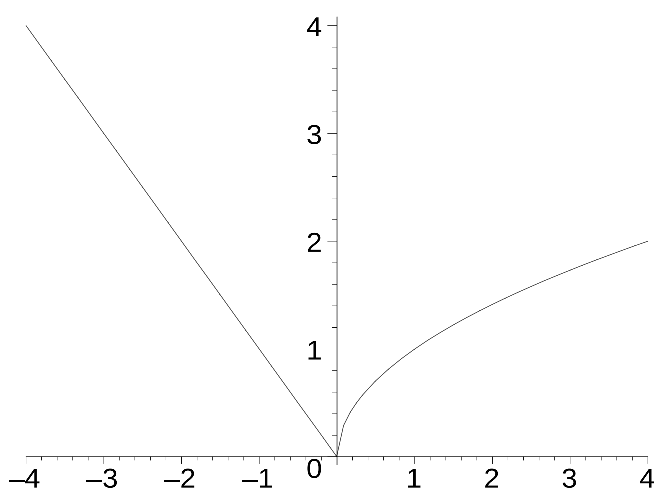
```
> F(x);
Error, (in F) cannot evaluate boolean: x < 0
```

The same error is reported when the function value **F(x)** is used in a **plot** command.

```
> plot( F(x), x=-4..4 );
Error, (in F) cannot evaluate boolean: x < 0
```

A nice plot is obtained when only the function name is used.

```
> plot( F, -4..4 );
```



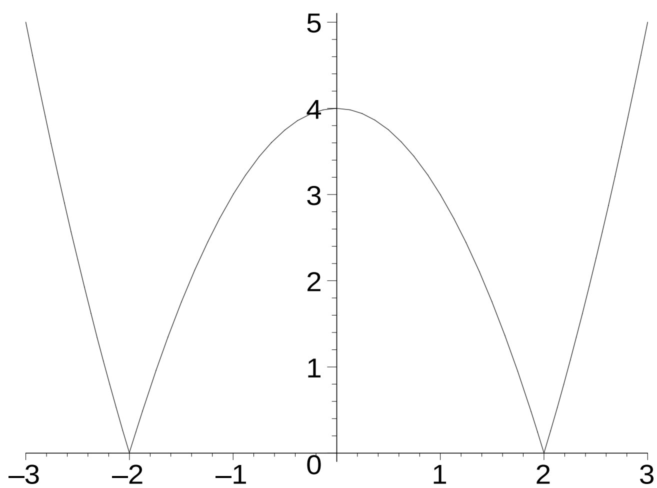
The **unapply** command provides a second way to define a Maple function.

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

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

Notice that the output uses the same arrow notation as was used to define **f** above.

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



E Lists and Sets

E.1 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 := { S, L };
```

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

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

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

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. (Neither should it be surprising to know that the order in which Maple displays the elements of a set can change from one session to another.)

E.2 Creating and Plotting 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 command (**$**). 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 ) ];
```

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

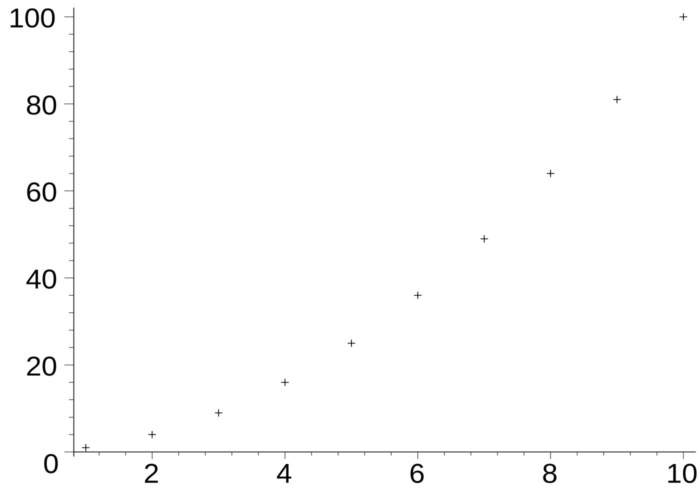
This list could also be created using **$** as

```
>  [ 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);
```

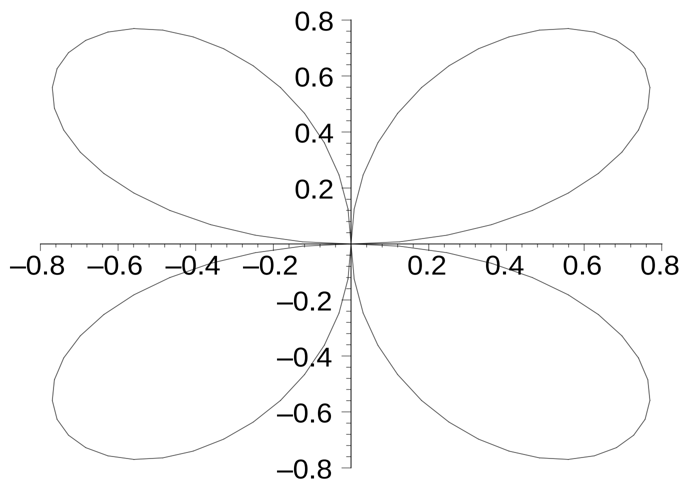


A list of 101 points on the polar curve $r = \sin(2\,t)$ can be constructed and displayed as follows. The list is not displayed as it is quite lengthy. Because the list elements include both the x- and y-coordinates, the **plot** command can be used; **listplot** can also be used to generate essentially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

```
> r := t -> sin(2*t);
```

$$r := t \to \sin(2\,t)$$

```
> 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 either of the following variations of the **plot** command. The output of these commands is suppressed by terminating the commands with a colon.

The first shows how to plot a polar function by specifying only the radius function and the range for the polar angle.

```
> plot( r(t), t = 0 .. 2*Pi, coords=polar ):
```

The second shows how the function can be displayed as a parametric curve.

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

E.3 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. 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 ];
```

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

A floating-point approximation to this point is

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

$$[.7639707480,\ .4848305795]$$

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

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

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

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 five-digit floating point approximations to the x-coordinates of each of these points can be obtained with

```
> seq( evalf[5](p[1]), p = rose4[ 1 ..  10 ] );
```

0., .12508, .24673, .36160, .46662, .55902, .63648, .69719, .73988, .76398

Sets can also be manipulated using the standard set operators: **union**, **intersect**, and **minus**. Each command can be used either 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" };
```

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

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

```
> word union num;
```

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

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

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

```
> word intersect num;
```

$$\{\text{"two"}\}$$

```
> num minus word;
```

$$\{\text{"four"}, \text{"three"}, \text{"one"}\}$$

The quotes on the command in the prefix form of the commands are required. The prefix forms of **union** and **intersect** can accept any number of sets.

E.4 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 ) ];
```

$$sol := [-1.732050808, 1.732050808]$$

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

$$-1.732050808$$

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

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

$$polysol := \{0., -1.879385242, 1.532088886, .3472963553\}$$

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

```
> polysol[2];
```

$$-1.879385242$$

The largest element of a set can be found using

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

$$1.532088886$$

The absolute value of each solution can be obtained using

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

$$\{0., 1.532088886, .3472963553, 1.879385242\}$$

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

```
> polysol2 := convert( polysol, list );
```

$$polysol2 := [0., -1.879385242, 1.532088886, .3472963553]$$

```
> sort( polysol2 );
```

$$[-1.879385242, 0., .3472963553, 1.532088886]$$

To sort the solutions by the magnitude of the solutions, the following variation of **sort** in which a user-defined ordering is used

```
> sort( polysol2, (a,b) -> evalb( abs(a) < abs(b) ) );
```

$$[0., .3472963553, 1.532088886, -1.879385242]$$

To conclude this discussion, consider the system of equations that describes the set of all points on the unit circle 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 } ) ];
```

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

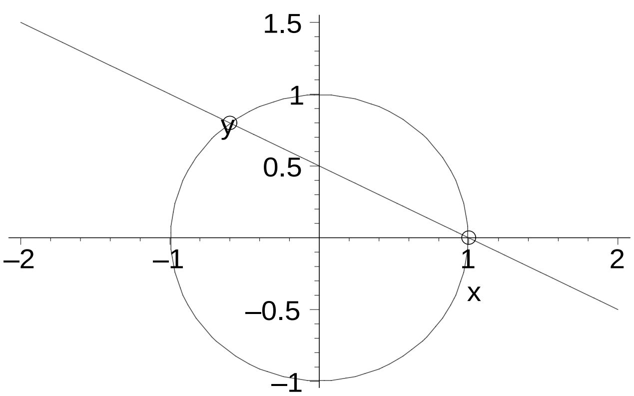
Each solution is a set of equations giving the x- and y-coordinates of a point on both curves. Note that the order of the equations within each solution is not consistent. These solutions can be converted to ordered pairs using

```
> 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 ] );
```



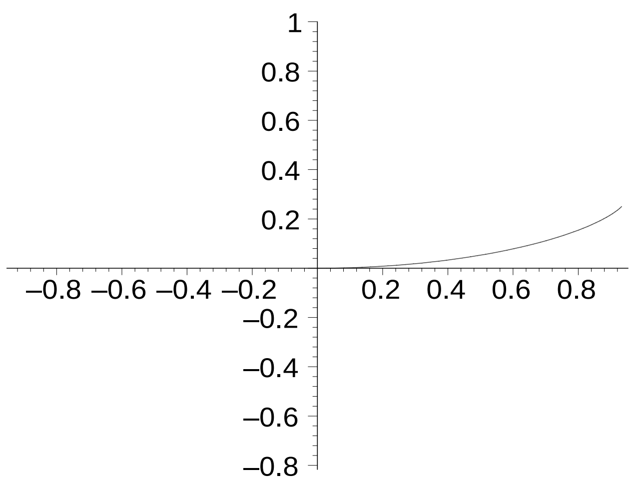
F Creating Animations

A Maple animation is formed when a list of Maple plots is displayed in sequence. 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 on top of 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.

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

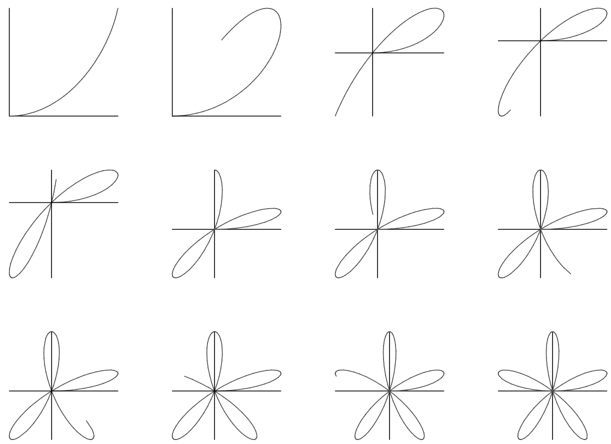


For a hardcopy such as this, it makes more sense to display all 12 frames of the animation as an array (or matrix) with three rows each containing four plots. The **matrix** command, from the **linalg** package, is used to create the 3 x 4 array of plots, then the **display** command is used to display 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 the point on the circle that makes a certain angle with the positive x-axis and the line segment from this point on the circle to the corresponding point on the sine curve for the same angle. 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) is 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 );
```

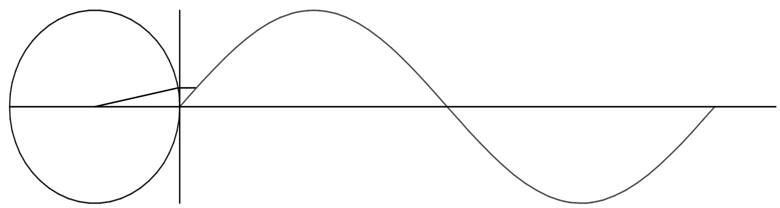
$$circ\_pt\_line := t \to \mathrm{plot}([[-1, 0], [\cos(t) - 1, \sin(t)], [t, \sin(t)]], color = black)$$

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

$$composite\_plot := t \to \mathrm{display}([p1, p2, \mathrm{circ\_pt\_line}(t)], view = [-2..7, -1..1], tickmarks = [0, 0], scaling = constrained)$$

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

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



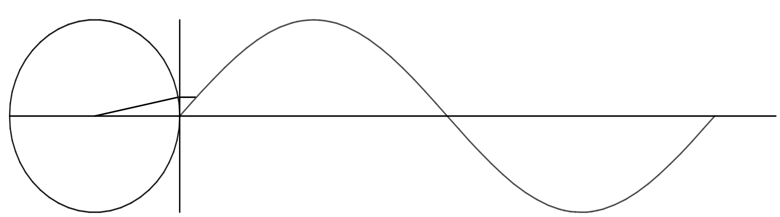
The list of 32 uniformly distributed angles between 0 and 2π is

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

$$angles := [\frac{1}{16}\pi, \frac{1}{8}\pi, \frac{3}{16}\pi, \frac{1}{4}\pi, \frac{5}{16}\pi, \frac{3}{8}\pi, \frac{7}{16}\pi, \frac{1}{2}\pi, \frac{9}{16}\pi, \frac{5}{8}\pi, \frac{11}{16}\pi, \frac{3}{4}\pi, \frac{13}{16}\pi, \frac{7}{8}\pi, \frac{15}{16}\pi, \pi, \frac{17}{16}\pi, \frac{9}{8}\pi, \frac{19}{16}\pi, \frac{5}{4}\pi, \frac{21}{16}\pi, \frac{11}{8}\pi, \frac{23}{16}\pi, \frac{3}{2}\pi, \frac{25}{16}\pi, \frac{13}{8}\pi, \frac{27}{16}\pi, \frac{7}{4}\pi, \frac{29}{16}\pi, \frac{15}{8}\pi, \frac{31}{16}\pi, 2\pi]$$

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 );
```



G 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 );
```

$$\frac{1}{2}\sqrt{2}$$

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 ];
```

$$\sin_{1/4\,\pi}$$

```
> exp({3+x}^2);
```

$$e^{(\{3+x\}^2)}$$

H Common Problems and How to Fix Them

H.1 Using a Command Before Loading It into 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 );
```

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

H.2 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 to extend or customize the functionality of some of Maple's built-in commands.

I Maple and Calculus

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

I.1 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 **a = infinity**.

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

$$e^x$$

Right- and left-hand limits are returned when the (optional) third argument is **right** and **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 );
```

$$\infty$$

The inert form of the limit command, **Limit**, can be used to produce very attractive mathematical displays.

```
> Limit( (x^2-4)/(x-2), x=2 ):
> % = value( % );
```

$$\lim_{x \to 2} \frac{x^2 - 4}{x - 2} = 4$$

I.2 Difference Quotients and Derivatives

```
> restart;
```

Given a function f, the slope of the secant lines through (x, f(x)) and ($x + h$, f($x + h$)) is a 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] );
```

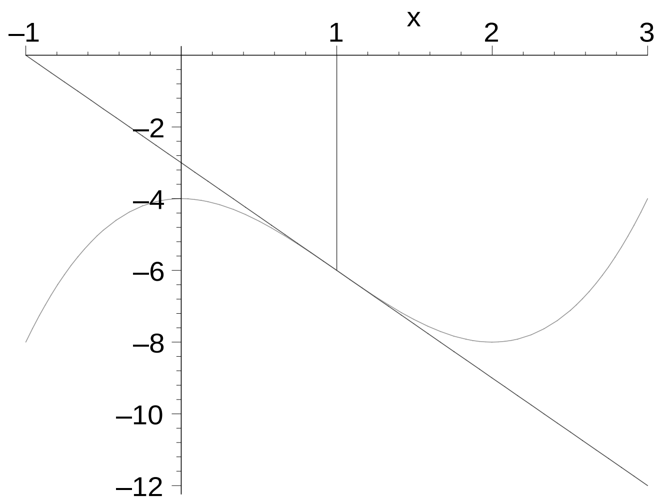
$$3\,x^2 + 3\,x\,h + h^2 - 6\,x - 3\,h$$

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

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

The **showtangent** command from the **student** package provides a simple means to display a function and the tangent line at a point.

```
> with( student ):
> showtangent( f(x), x=1, x=-1..3 );
```



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.

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

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

Implicit differentiation can be done in the same manner, assuming that all functional dependencies are explicitly noted in the equation. The following example demonstrates that the implicit derivative is the same as would be obtained explicitly.

We begin with the implicitly defined function

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

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

Because this equation is quadratic in y, it is expected that there will be two values of y for each value of x. The quadratic formula can be used to obtain explicit formulas for y in terms of x.

```
> expl_eq := solve( impl_eq, y );
```

$$expl\_eq := \sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x},\ -\sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x}$$

The derivative of each branch is found as above.

```
> expl_dydx[1] := simplify( diff( expl_eq[1], x ) );
```

$$expl\_dydx_1 := -\frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x}}$$

```
> expl_dydx[2] := simplify( diff( expl_eq[2], x ) );
```

$$expl\_dydx_2 := \frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x}}$$

The same derivatives can be found by rewriting the implicit differentiation to explicitly note the dependence of y on x and differentiating the result with respect to x.

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

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

```
> diff( %, x );
```

$$4\,(x-1)^3 = 2\,x - 2\,\mathrm{y}(x)\,(\tfrac{\partial}{\partial x}\,\mathrm{y}(x))$$

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

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

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

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

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

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

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

```
> simplify( eval( impl_dydx, y(x)=expl_eq[1] ) );
```

$$-\frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x}} = -\frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x}}$$

```
> simplify( eval( impl_dydx, y(x)=expl_eq[2] ) );
```

$$\frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x}} = \frac{2\,x^3 - 6\,x^2 + 5\,x - 2}{\sqrt{-1 - x^4 + 4\,x^3 - 5\,x^2 + 4\,x}}$$

Note that Maple typesets all derivatives — ordinary or partial — with ”curly d” notation. This convention, which cannot be changed, should be explained to students. In fact, it is rather easy to use this as a learning opportunity and to introduce the concept of partial derivative in Calculus I.

I.3 Applications of Derivatives

```
> restart;
```

Linearization of a Function

The linearization of a function f at $x = a$ is L(x) = f(a) + f'(a) ($x - a$). To create this function in Maple when

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

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

and

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

$$a := 0$$

use either of the following

```
> L1 := x -> f(a) + D(f)(a)*(x-a);
```

$$L1 := x \to \mathrm{f}(a) + \mathrm{D}(f)(a)\,(x-a)$$

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

$$L2 := x \to 1 + \frac{1}{2}\,x$$

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 is changed, then the function **L1** will reflect those changes as well. The function **L2**, however, uses the values of f and a at the time the function is created and is unaffected by subsequent changes. This is illustrated in the following examples.

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

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

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

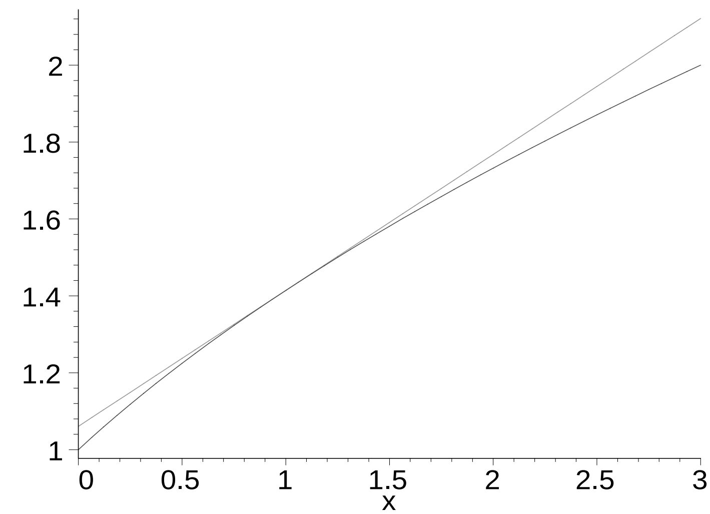
$$a := 1$$

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

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

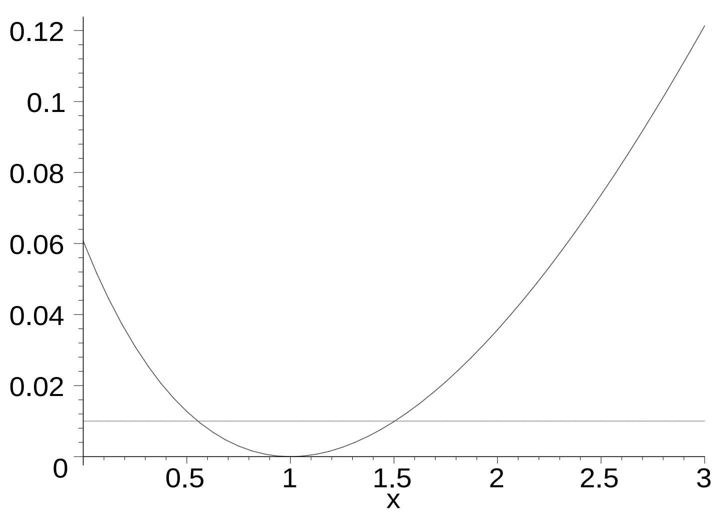
The function and its linearization at a are graphed below.

```
> plot( [ f(x), L1(x) ], x = 0 .. 3 );
```



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)-L1(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 a point close to (0.56, 0.01). Repeating this process for the rightmost intersection gives (1.50, 0.01). This gives the estimate δ = min(1-0.56, 1.50-1) = 0.44.

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

```
> solve( abs( f(x)-L1(x) ) < epsilon, x );
```

$$\mathrm{RealRange}(\mathrm{Open}(.5526014252), \mathrm{Open}(1.503967117))$$

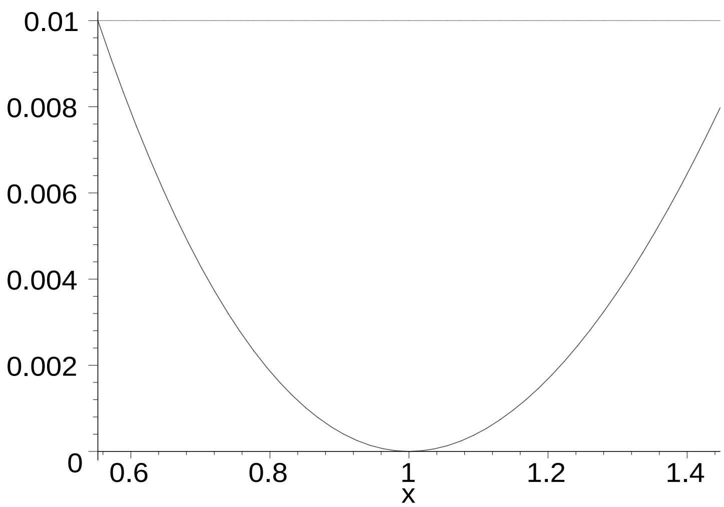
This yields

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

$$\delta := .4473985748$$

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

```
> plot( [ abs( f(x)-L1(x) ), epsilon ], x = 1-delta .. 1+delta );
```




```
> restart;
```

Newton's Method

Newton's Method can be implemented in Maple to find an approximation to the smallest solution to $\mathrm{f}(x) = 0$ with

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

$$f := x \to \cos(5\,x) - x$$

to five decimal places by defining the auxiliary function

```
> g := unapply( x - f(x)/D(f)(x), x );
```

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

and the initial guess (use a graph, if necessary, to obtain a starting value)

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

$$x0 := -1.$$

The next approximation is

```
> x1 := g( x0 );
```

$$x1 := -.7784735010$$

Continuing in the same manner until the iterates agree to five decimal places

```
> x2 := g( x1 );
```

$$x2 := -.7677474241$$

```
> x3 := g( x2 );
```

$$x3 := -.7674935683$$

```
> x4 := g( x3 );
```

$$x4 := -.7674934212$$

Thus, $x = -.76749$ is the approximate value of the smallest root. 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** and the concatenation operator (——) for more information.)

```
> for n from 2 while abs( x||(n-1) - x||(n-2) ) >= 10^(-6) do
>   x||n := g( x||(n-1) );
> end do;
```

$$x2 := -.7677474241$$
$$x3 := -.7674935683$$
$$x4 := -.7674934212$$

I.4 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 y'(x) = $5\, e^{(-3\,x)}$, 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;
```

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

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

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

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

and solve for C

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

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

The solution to the initial value problem is

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

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

Note that the same result is obtained in one step from the **dsolve** command (which will be discussed in more detail shortly).

```
> dsolve( { diff( y(x), x ) = 5*exp(-3*x), y(0) = -10 }, y(x) );
```

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

The **student** package contains a number of commands for the visualization and estimation of Riemann sums.

```
> 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*]

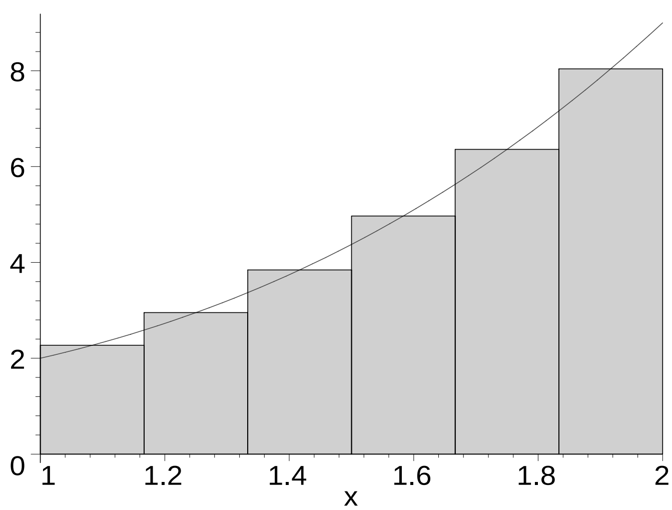
To illustrate, consider the problem of approximating the value of $\int_1^2 x^3 + 1\,dx$. Let

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

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

The midpoint approximation to this integral with 6 subdivisions can be visualized with

```
> middlebox( f, x = 1 .. 2, 6 );
```



The numerical approximation to the integral, the area of the six rectangles, is

```
> middlesum( f, x = 1 .. 2, 6 ):
> % = evalf( % );
```

$$\frac{1}{6}\left(\sum_{i=0}^{5}((\frac{13}{12}+\frac{1}{6}i)^3+1)\right)=4.739583334$$

For right-hand sums use **rightbox** and **rightsum**. For left-hand sums, **leftbox** and **leftsum**. For the Trapezoidal and Simpson's Rules, **trapezoid** and **simpson** produce numerical approximations; there are no built-in commands to produce the illustrate these methods graphically. For example,

```
> evalf( trapezoid( f, x = 1 .. 2, 6 ) );
```

$$4.770833334$$

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 the learning of techniques such as substitution and integration by parts. To prevent Maple from evaluating an integral – definite or indefinite – use **Int**, the inert form of **int**. (The **value** command is used to force the symbolic evaluation of an inert command; **evalf** is used for numerical evaluation.) A substitution is carried out with the **changevar** command.

```
> ex1 := Int( x * sqrt( 3*x+1 ), x = 0 .. 3 );
```

$$ex1 := \int_0^3 x\sqrt{3x+1}\,dx$$

```
> changevar( 3*x+1 = u, ex1 );
```

$$\int_1^{10}\frac{1}{3}(-\frac{1}{3}+\frac{1}{3}u)\sqrt{u}\,du$$

```
> simplify( % );
```

$$\frac{1}{9}\int_1^{10}(-1+u)\sqrt{u}\,du$$

```
> ex1 = %;`` = value( %% );
```

$$\int_0^3 x\sqrt{3x+1}\,dx = \frac{1}{9}\int_1^{10}(-1+u)\sqrt{u}\,du$$

$$= \frac{100}{27}\sqrt{10}+\frac{4}{135}$$

To check this answer, compare it with the Maple-generated value of the original integral

```
> ex1 = value( ex1 );
```

$$\int_0^3 x\,\sqrt{3\,x+1}\,dx = \frac{100}{27}\,\sqrt{10} + \frac{4}{135}$$

The indefinite integral

```
> ex2 := Int( x*sin(a*x), x );
```

$$\mathit{ex2} := \int x\sin(a\,x)\,dx$$

can be evaluated using integration by parts. The value of u must be specified in the second argument of the **intparts** command.

```
> ex2 = intparts( ex2, x );
```

$$\int x\sin(a\,x)\,dx = -\frac{x\cos(a\,x)}{a} - \int -\frac{\cos(a\,x)}{a}\,dx$$

This integral is now easily evaluated; the result is

```
> ex2 = value( rhs(%) ) + C;
```

$$\int x\sin(a\,x)\,dx = -\frac{x\cos(a\,x)}{a} + \frac{\sin(a\,x)}{a^2} + C$$

which is seen to be equivalent to the Maple-generated value of the original integral

```
> ex2 = value( ex2 ) + C;
```

$$\int x\sin(a\,x)\,dx = \frac{\sin(a\,x) - a\,x\cos(a\,x)}{a^2} + C$$

I.5 Applications of Integrals

```
> restart;
```

Arclength of a Smooth Curve

The definite integral for the arclength of a smooth curve is usually easy to setup but difficult to evaluate.

```
> f := 4*sqrt(2)/3 * x^(3/2) - 1;
```

$$f := \frac{4}{3}\sqrt{2}\,x^{(3/2)} - 1$$

```
> arclength := Int( sqrt( 1 + diff(f,x)^2 ), x=0..1 );
```

$$\mathit{arclength} := \int_0^1 \sqrt{1+8\,x}\,dx$$

```
> arclength = value( arclength );
```

$$\int_0^1 \sqrt{1+8\,x}\,dx = \frac{13}{6}$$

Or, if the curve is entered in Maple as a function

```
> F := x -> 4*sqrt(2)/3 * x^(3/2) - 1;
```

$$F := x \to \frac{4}{3}\sqrt{2}\,x^{(3/2)} - 1$$

```
> int( sqrt( 1 + D(F)(x)^2 ), x=0..1 );
```

$$\frac{13}{6}$$

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 := exp(-x);
```

$$f := e^{(-x)}$$

```
> int( sqrt( 1 + diff(f,x)^2 ), x=0..1 );
```

$$-\sqrt{1+e^{(-2)}} + \operatorname{arctanh}\Big(\frac{1}{\sqrt{1+e^{(-2)}}}\Big) + \sqrt{2} - \operatorname{arctanh}\Big(\frac{1}{2}\sqrt{2}\Big)$$

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

$$1.192701404$$

Note that the **evalf** command returns a numerical approximation to the exact symbolic value of the definite integral. To obtain this value as a numerical approximation to the definite integral, use the **evalf** command to evaluate an inert definite integral.

```
> evalf( Int( sqrt( 1 + diff(f,x)^2 ), x=0..1 ) );
```
$$1.192701402$$

```
> restart;
```

Improper Integrals

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 );
```
$$\frac{1}{2}\,\pi$$

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

```
> restart;
```

Multiple Integrals

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 );
```
$$-2+e$$

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 );
```
$$a\,b\,\pi$$

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

I.6 Differential Equations

```
> restart;
```

The **dsolve** command attempts to solve a differential equation or initial value problem.

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

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

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

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

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 conditions" into the solution, solve for **_C1**, and substitute the result back into the general solution.

```
> subs( { y(x) = 2, x=0 }, soln );
```

$$2 = 3\,\frac{1}{1 + 3\,e^0\,\_C1}$$

```
> const := isolate( %, _C1 );
```

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

```
> eval( soln, const );
```

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

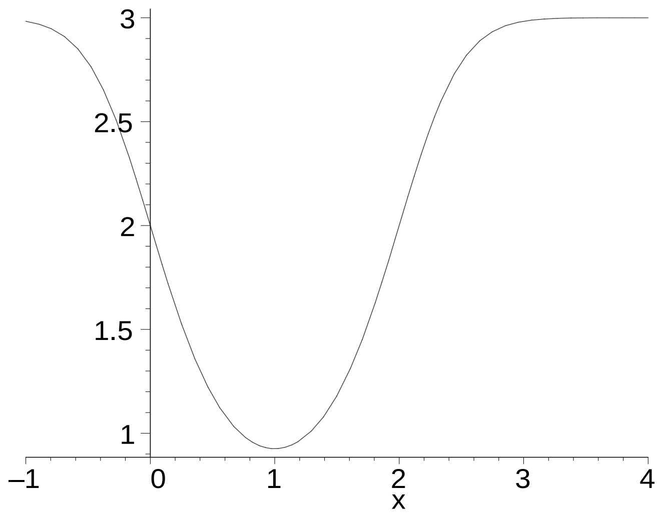
If the solution to an initial value problem is all that is needed, then the above process can be streamlined to a single **dsolve** command in which the first argument is a set containing the differential equation and the initial condition.

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

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

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(%), x=-1..4 );
```



Many initial value problems cannot be solved explicitly, or implicitly. The addition of the optional argument **numeric** to the **dsolve** command instructs Maple to return a procedure that computes a numerical approximation of the solution at a specified point.

```
> with( plots ):
Warning, the name changecoords has been redefined
> S := dsolve( { ode, y(0)=2 }, y(x), numeric );
```

$$S := \textbf{proc}(\mathit{rkf45\_x})\ \ldots\ \textbf{end proc}$$

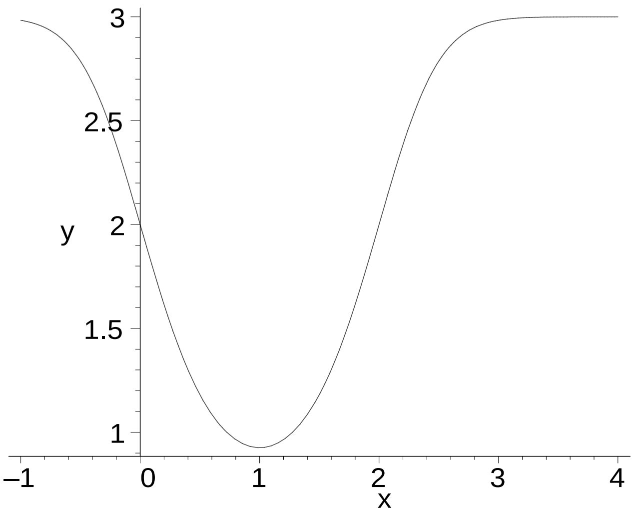
The unusual output of this **dsolve** command is simply a reminder that S is a Maple procedure, *i.e.*, a function. When this function is evaluated the result is an approximate value of the solution to the initial value problem at the specified value of the independent variable.

```
> S(1);
```

$$[x = 1., \mathrm{y}(x) = .925685934862923609]$$

Another use for a numerical solution of an IVP is to use the **odeplot** command, from the **plots** package, to create a plot of a numerical solution to the initial value problem.

```
> odeplot( S, [ x, y(x) ], -1..4 );
```

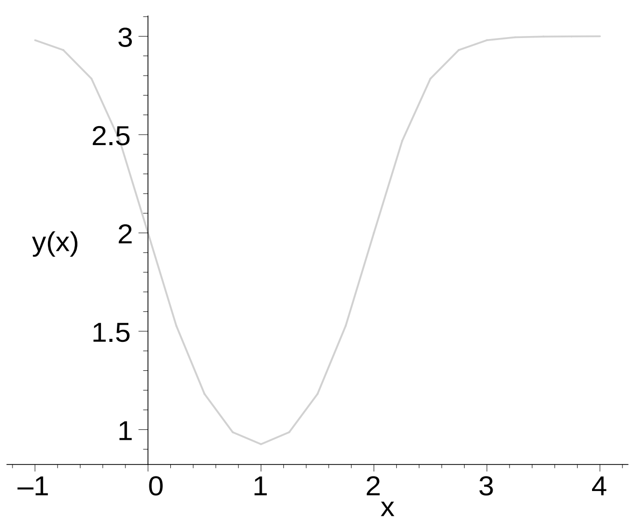


The **DEtools** package contains a number of useful commands for working with differential equations.

```
> with( DEtools ):
```

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 );
```

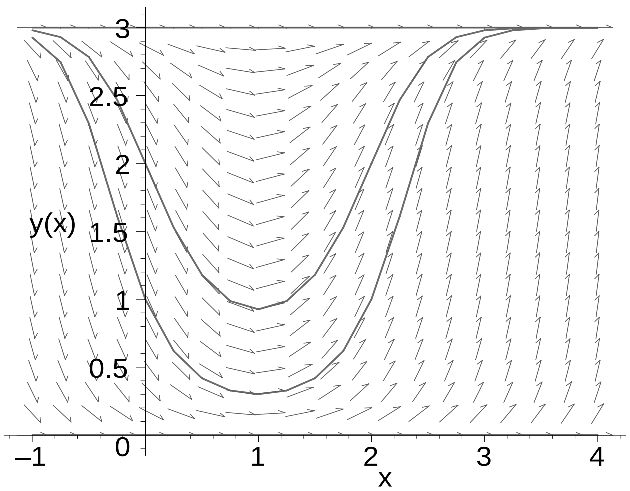


Changing the last argument to **arrows=thin** will include the direction field. Adding additional ordered pairs to the list of initial conditions adds additional solution curves to the plot.

```
> 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 );
```



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.

J Concluding Remarks

This concludes our brief tutorial. At this point you should be able to successfully complete many of the Technology Exercises and Student Projects.

For additional online tutorial information do not forget about the **New User's Tour**.