Preface

MAPLE is a computer algebra system which, thanks to an extensive library of sophisticated functions, enables both numerical and formal computations to be performed. Until recently, such systems were only available to professional users with access to mainframe computers, but the rapid improvement in the performance of personal computers (speed, memory) now makes them accessible to the majority of users. The latest versions of MAPLE belong to this new generation of systems, allowing a growing audience of users to become familiar with computer algebra.

This work does not set out to describe all the possibilities of MAPLE in an exhaustive manner; there is already a great deal of such documentation, including extensive online help. However, these technical manuals provide a mass of information which is not always of great help to a beginner in computer algebra who is looking for a quick solution to a problem in his own speciality: mathematics, physics, chemistry, etc.

This book has been designed so that a scientist who wishes to use MAPLE can find the information he requires quickly. It is divided into chapters which are largely independent, each one being devoted to a separate subject (graphics, differential equations, integration, polynomials, linear algebra, ...), enabling each user to concentrate on the functions he really needs. In each chapter, deliberately simple examples have been given in order to fully illustrate the syntax used.

The authors have gone to great lengths to give numerous examples of typical incorrect uses of functions in order to help the reader decode and interpret error messages which he may encounter when using MAPLE later on.

How to use this book:

- Chapter 1 is a “guided tour” enabling the reader to catch a glimpse of the different possibilities of MAPLE.
- Chapter 2 presents the basic functions of MAPLE, and it is strongly recommended that Sections 2.1, 2.2 and 2.3 of this chapter be read in order to fully benefit from the rest of the work. In particular, the reader should pay special attention to Section 2.3, which is devoted to evaluation rules, which play a fundamental role in MAPLE.
- The following chapters are to a large extent independent and can be approached directly according to the areas of interest of the user. However, following these guidelines is recommended.
    - ∗ Chapter 9 on differential equations is best read after studying Chapter 5 on two-dimensional graphics, and possibly Section 7.2 devoted to derivatives.
    - ∗ Chapters 15 to 18, dealing with linear algebra, form a whole which requires a detailed study of Section 15.4, which is devoted to evaluation problems in linear algebra.
    - ∗ Chapter 13, on polynomials with irrational coefficients requires a detailed study of Chapter 12.
    - ∗ Chapter 22 studies the `subs` et `map` functions which enable the user to work more cleverly with MAPLE, but for a thorough study of this chapter it is necessary to have a good knowledge of Chapter 21, which describes the MAPLE objects.
    - ∗ Chapters 23 and 24 introduce some programming notions: loops, branches, and procedures. These tools become indispensable when one wishes to use MAPLE to perform the same series of computations for several different cases. They are presented in an elementary way with simple examples.

It is impossible in a single book to give an exhaustive introduction to all of the MAPLE functions. If the basic functions are described in detail with a precise syntax, the others are presented with the help of sufficiently explicit examples, together with comments enabling their use to be understood. The index, which is very detailed, will help the user to find them.

This book is mainly designed for students, teachers, engineers and researchers who aim to quickly master a software system for formal calculations, which is becoming more and more indispensable for their activity, but also for anyone wishing to solve a problem or test a conjecture for which the calculations appear at first dissuasive.

However, the user should keep in mind that MAPLE is a product in constant development with a few imperfections. He should therefore cast a critical eye on the results obtained and not hesitate to try to check them by different methods.

11. Three-Dimensional Graphics

11.1 Surfaces Defined by an Equation $z = f(x, y)$

In this part, we study surfaces defined by an equation $z = f(x, y)$. In MAPLE, such a surface can be defined either by an expression depending on the variables `x` and `y` or by a function in two variables.

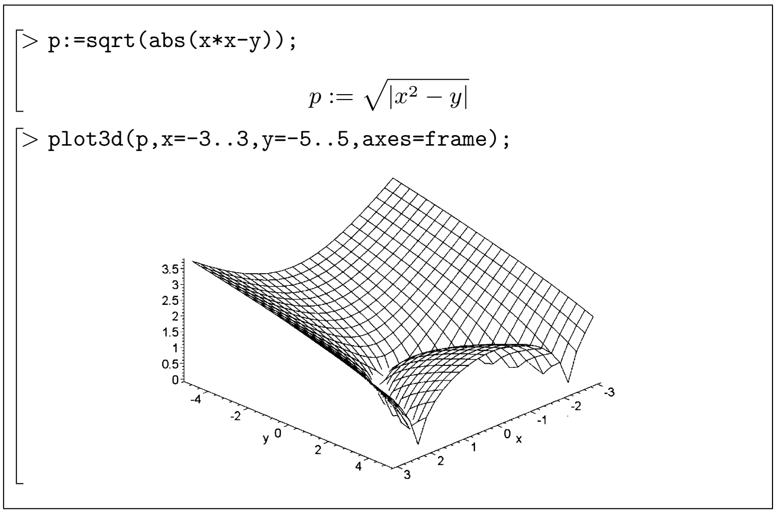
11.1.1 Plot of a Surface Defined by an Expression

If `p` is an expression in the free variables `x` and `y`, and if `a`, `b`, `c` and `d` are four numerical values such that `a<b` and `c<d`, then `plot3d(p,x=a..b,y=c..d)` plots the surface defined by

$$= \{(x, y, p) \mid a \quad x \quad b \text{ and } c \quad y \quad d\}$$

Since the function `plot3d` doesn't plot any axes by default, using `axes=frame` or `axes=boxed` as an option, which displays the axes, is recommended. This way plots are produced are more easily understood.

Ex. 1



```
> p:=sqrt(abs(x*x-y));
```

$$p := \sqrt{|x^2 - y|}$$

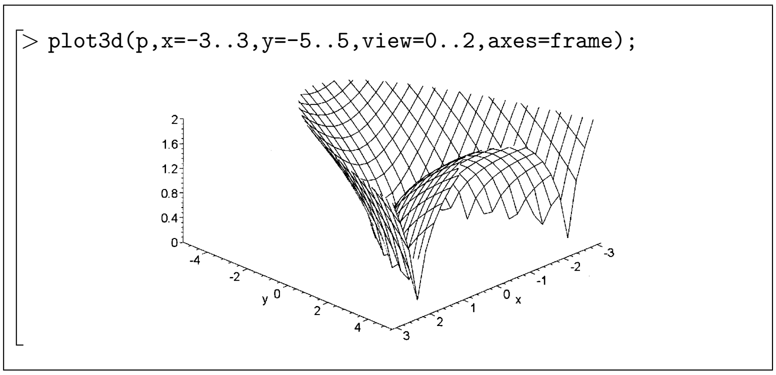
```
> plot3d(p,x=-3..3,y=-5..5,axes=frame);
```

By default, the plot is included in the worksheet and its size can be adjusted with the help of the mouse. However, if the option **Plot Display→Window** in the **Options** menu has been selected before the execution of `plot`, the plot is displayed alone in another window, where it can be printed without the other computations.

As the previous example shows, MAPLE automatically adjusts the scale of `z` in order to represent all the points of the surface , the plot is then displayed "full page". To restrict the range of `z` to `[z1,z2]`, either to zoom in on a subset of the surface or because the suface "goes to infinity", use the option `view=z1..z2`.

Example of zooming in on a subset of the previous surface.

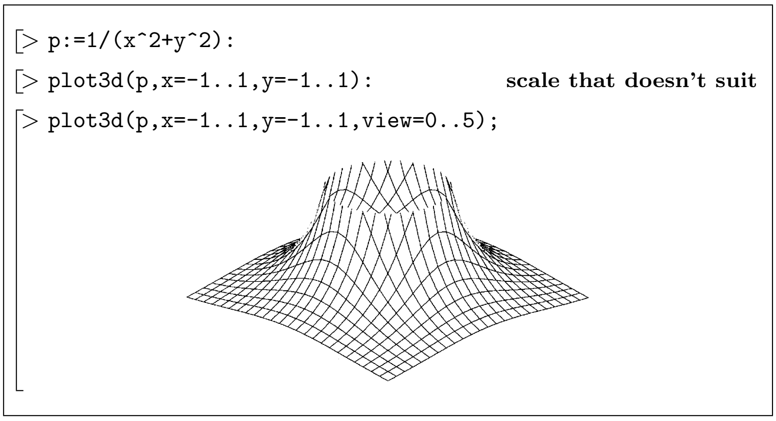
Ex. 2



```
> plot3d(p,x=-3..3,y=-5..5,view=0..2,axes=frame);
```

Example in which the use of `view` is essential since the surface is defined by an expression that has an infinite limit at a point in the domain.

Ex. 3



```
> p:=1/(x^2+y^2):
> plot3d(p,x=-1..1,y=-1..1):
> plot3d(p,x=-1..1,y=-1..1,view=0..5);
```

scale that doesn't suit

Warning! If either of the variables `x` or `y` isn't free when MAPLE evaluates `plot3d(p,x=a..b,y=c..d)`, it returns the following error message:

Ex. 4

```
> x:=t^2+1:plot3d(p,x=-1..1,y=-1..1);
  Error, (in plot3d/expression) bad range arguments,
  t^2+1=-1..1, y=-1..1
```

Warning!

- If one forgets the identifiers `x` and `y` appearing as second and third parameters, MAPLE returns a message saying that the plot is empty without further explanation.
- If only one of the identifier `x` and `y` is forgotten, MAPLE returns an error message saying that the arguments are not suitable.

Ex. 5

```
> plot3d(1/(x^2+y^2),-1..1,-1..1);
  Plotting error, empty plot
> plot3d(1/(x^2+y^2),x=-1..1,-1..1);
  Error, (in plot3d/expression) bad range arguments,
  x=-1..1,-1..1
```

11.1.2 Plot of a Surface Defined by a Function

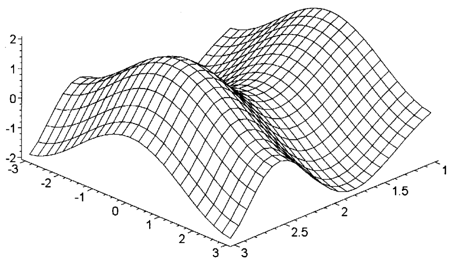
If `f` is a function in two variables (or a procedure) and if `a`, `b`, `c` and `d` are four real numbers such that `a<b` and `b<d`, then `plot3d(f(x,y),x=a..b,y=c..d)` or `plot3d(f,a..b,c..d)` plots the surface defined by $z = f(x, y)$, for x ranging from a to b and y ranging from c to d. The range of z can be restricted to the interval $[z1, z2]$ by using the option `view=z1..z2`.

Ex. 6

```
> f:=(x,y)->cos(x*x)+cos(y);
```

$$f := (x, y) \to \cos(x^2) + \cos(y)$$

```
> plot3d(f,1..3,-3..3,axes=frame);
```



11.1.3 Simultaneous Plot of Several Surfaces

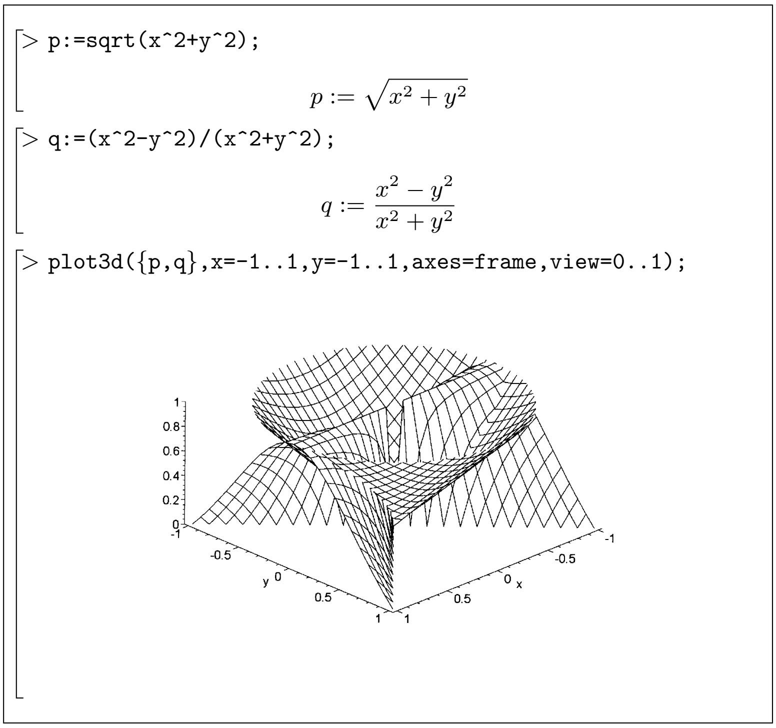
If `p` and `q` are two expressions in the free variables `x` and `y`, and if `a`, `b`, `c` and `d` are numerical values such that `a<b` and `b<d`, then the evaluation of `plot3d({p,q},x=a..b,y=c..d)` plots the surfaces $_1$ and $_2$ defined by

$$\begin{aligned} {}_1 &= \{(x,y,p) \mid a \quad x \quad b \text{ and } c \quad y \quad d\} \\ {}_2 &= \{(x,y,q) \mid a \quad x \quad b \text{ and } c \quad y \quad d\} \end{aligned}$$

Warning ! Do not forget the braces around `p` and `q`, and do not replace them with square brackets, which have a totally different meaning as explained in Section 11.3, p. 189.

Example showing the simultaneous plot of a cone and of the classical Plücker's conoid.

Ex. 7



```
> p:=sqrt(x^2+y^2);
```

$$p := \sqrt{x^2+y^2}$$

```
> q:=(x^2-y^2)/(x^2+y^2);
```

$$q := \frac{x^2-y^2}{x^2+y^2}$$

```
> plot3d({p,q},x=-1..1,y=-1..1,axes=frame,view=0..1);
```

11.2 The Environment of `plot3d`

By default, a plot is included in the worksheet. However, if the option **Plot Display→Window** of the **Options** menu has been selected before the execution of `plot`, the plot is displayed alone in an external window. Several options which can be accessed through menus, most of them described below, can be used to modify the appearance of the plot; they can be directly accessed when the plot is displayed in an external window. If the drawing is included in the worksheet, one must click onto the plot to access them.

11.2.1 The Menu of `plot3d` in Windows

In graphic mode, there is a menu bar, a tool bar and a context bar that contain, among other things, the button $\boxed{\text{R}}$, which replots the surface after any style modification.

The Menu Bar

In graphic mode, the most useful options of the menu bar are:

File same functions as in Section 5.2, p. 75.

Edit to copy the plot into the clipboard (*copy*) in order to transfer it to another Windows compatible application.

View to validate or suppress the display of the tool bar (*Toolbar*) or of the status bar (*Statusbar*).

Style to choose among the following plot styles:

Patch colored surface and grid obtained by joining the computed points.

Patch w/o grid colored surface, without grid.

Patch and contour colored surface, plot of the level curves in z.

Hidden line grid obtained by joining the computed points, hidden lines not plotted.

Contour plot of the level lines in z.

Wireframe grid obtained by joining the computed points, hidden lines included.

Point plot of the computed points only.

Style to choose the **Symbol** being used when the surface is plotted with points, which can be: **Cross**, **Diamond**, **Point**, **Circle**, ...

to choose the **Line Style** being used to represent the curves, which can be: continuous (**Solid**), dashes (**Dash**), ...

to choose the **Grid Style**, either quadrilaterals (**Grid Half**) or quadrilaterals with a diagonal (**Grid Full**).

to choose the **line width**, which can be **thin**, **medium**, **thick** or the **default** width.

Color to choose, in **Patch** style, the distribution mode of the colors

XYZ colors varying in function of **X**, **Y** and **Z**.

XY colors varying in function of **X** and **Y**.

Z colors varying in function of **Z** alone.

Z(**Hue**) as the previous one, but with stronger variations.

Z(**Greyscale**) grey tinted surface with shades varying in function of **Z**.

NoColoring white surface.

to choose a lighting of the surface or not and to specify its origin, it can be **No Lighting**, **light scheme1**, **light scheme2**, . . .

Axes to choose the kind and the position of the axes.

Boxed axes forming a parallelepiped around the surface.

Framed axes plotted at the edges of the reference parallelepiped.

Normal axes through, in general, the origin.

None no axes plotted.

Projection to choose the type of perspective used: **No Perspective**, **Near Perspective**, **Medium Perspective** or **Far Perspective**.

to choose among an orthonormal frame (**Constrained**) or a plot using the window best (**Unconstrained**).

The Tool Bar and the Context Bar

The icons of the tool bar and the context bar allow one, by clicking upon them with the mouse's left button, to directly activate some options of the previous menu. Every icon is suggestive enough and a message describing its use can be displayed in the status bar by clicking upon the mouse's left button while the cursor stands above this icon. The choice will only be activated if the mouse's button let up inside the icon.

Orientation of the Reference Parallelepiped

When one clicks on the plot window with the mouse, the plotted surface disappears and is replaced by a “parallelepiped” whose sides are parallel to the axes while the values of θ and , which determine the angle from which the surface is viewed, are displayed on the left side of the context bar.

The orientation of the reference parallelepiped can then be manually modified, which means modifying the position of the observer, by sliding the mouse’s pointer while the mouse’s left button is depressed. During this process, the values of θ and , which determine the position of the observer, are continuously displayed in the context bar. Once an orientation has been chosen, one has only to depress the $\boxed{\text{ENTER}}$ key or to click upon the icon $\boxed{\text{R}}$ to obtain the new plot.

Note: The values of θ and can also be directly modified either by entering numerical values (in degrees) or by using the variation arrows.

11.2.2 The Options of `plot3d`

Some options can be added when calling `plot3d` which can be used to specify a presentation style or to refine the plot. Several options may be used, separated with commas. As for the function `plot`, these options have to be put after the three first parameters, which are the expression or the function and the intervals in `x` and in `y`.

Presentation Options Corresponding to Some Menu Options

Some options of `plot3d` can be used to directly obtain some options of the previous menu.

`style` used in the form `style=hidden`, `style=contour`, `style=patch`, `style=patchnogrid`, `style=patchcontour`, `style=wireframe` or
`style=point`, according to the plot style one wishes. The default style is `hidden`.

`thickness` used in the form `thickness=n` where `n` equals `0`, `1`, `2` or `3`, according to the line width one wishes.

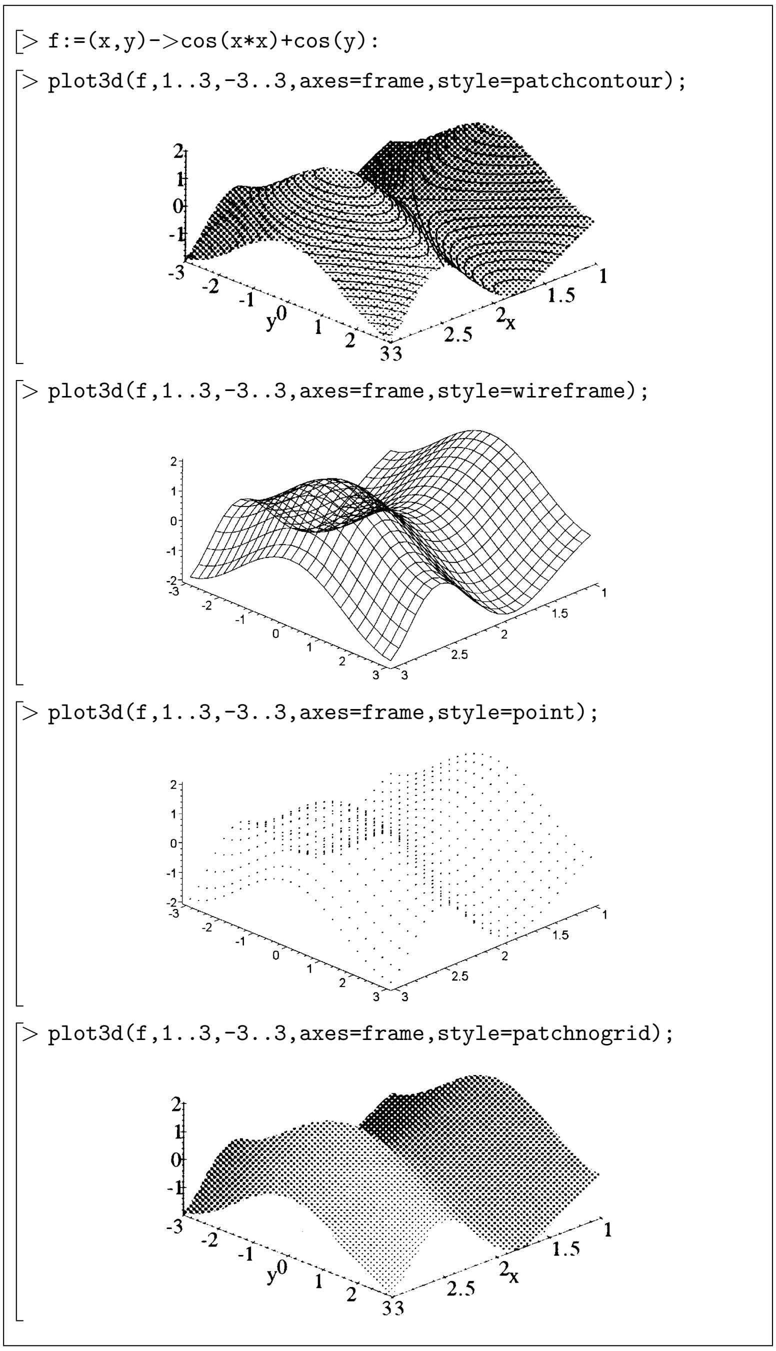
`axes` used in the form `axes=boxed`, `axes=framed`, `axes=normal`, or `axes=none` according to the kind of axes one wishes.

`scaling` `scaling=constrained` or `scaling=unconstrained` which chooses between a representation with orthonormal axes and a representation that uses the window best.

`shading` used in the form `shading=s`, where `s` is equal to `xyz`, `xy`, `z`, `zhue`, `zgreyscale`, `none`, according to the distribution of shades one wishes.

Comparison between the different plot styles for the surface of Example 6, p. 181

Ex. 8



```
> f:=(x,y)->cos(x*x)+cos(y):
> plot3d(f,1..3,-3..3,axes=frame,style=patchcontour);
> plot3d(f,1..3,-3..3,axes=frame,style=wireframe);
> plot3d(f,1..3,-3..3,axes=frame,style=point);
> plot3d(f,1..3,-3..3,axes=frame,style=patchnogrid);
```

Presentation Options Not Corresponding to a Menu Option

tickmarks tickmarks=[l,m,n] is used to display about l values on the x axis, m values on the y axis and n values on the z axis.

orientation orientation=[m,n] is used to choose the values in degrees of *theta* (longitude) and of *phi* (latitude) which determine the angle from which the surface is viewed. By default we have *theta* = *phi* = 45°.

view view=z1..z2 is used to restrict the values of z to $[z1, z2]$. view=[x1..x2,y1..y2,z1..z2] is used to restrict the values of the three coordinates.

labels labels=[x,y,z] is used to label the axes.

title title='surface name' is used to display a title.

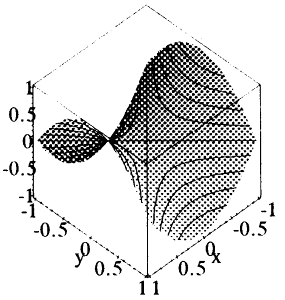
Examples of plots of the same surface viewed from different view points.

```
> plots[setoptions3d](scaling=constrained,
  axes=boxed,style=patchcontour);
```

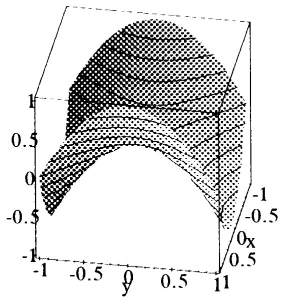
same use as in Section 5.2, p. 81

```
> p:=x^2-y^2 : plot3d(p,x=-1..1,y=-1..1);
```

Ex. 9

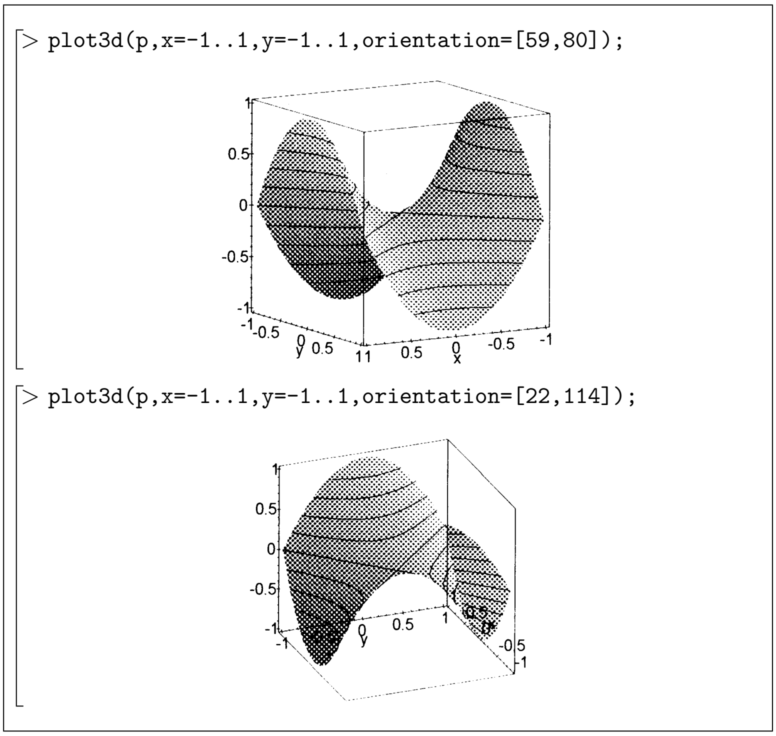


```
> plot3d(p,x=-1..1,y=-1..1,orientation=[10,60]);
```



The previous surface viewed from other angles

Ex.10



```
> plot3d(p,x=-1..1,y=-1..1,orientation=[59,80]);
```

```
> plot3d(p,x=-1..1,y=-1..1,orientation=[22,114]);
```

The Options `numpoints` and `grid`

When evaluating `plot3d(p,x=a..b,y=c..d)`, MAPLE computes `p` for a set of values (x, y) forming a grid of points in the rectangle $[a, b] \times [c, d]$ whose abscissas and ordinates are regularly spaced. By default, this grid is square and contains $25^2 = 625$ points. The two following options can be used to refine this grid.

`grid` with `grid=[m,n]`, the grid contains points along the axis of abscissas and n points along the axis of ordinates.

`numpoints` with `numpoints=n`, MAPLE uses a grid of about $\sqrt{n} \times \sqrt{n}$.

Warning ! Even more than for the function `plot`, these options should be managed according to the capabilities of the system and the time at one's disposal.

11.3 Surface Patches Parametrized in Cartesian Coordinates

This section concerns parametrized surface patches, i.e. surfaces comprised of points whose coordinates, (x, y, z), depend on two parameters. Such a parametric representation can be defined either by expressions depending on two variables or by functions or procedures of two variables.

Given four numerical values, a, b, c, d, such that `a<b` and `c<d`

- If `p`, `q` and `r` are expressions in the two free variables `u` and `v`, then `plot3d([p,q,r],u=a..b,v=c..d)` plots the surface which is defined parametrically by

$$= \{(p, q, r) \mid a \quad u \quad b \text{ and } c \quad v \quad d\}$$

- If `f`, `g` and `h` are functions (or procedures) of two variables, then the evaluation of `plot3d([f(u,v),g(u,v),h(u,v)],u=a..b,v=c..d)` or of `plot3d([f,g,h],a..b,c..d)` plots the surface which is defined parametrically by

$$= \{(f(u,v), g(u,v), \ (u,v)) \mid a \quad u \quad b \text{ and } c \quad v \quad d\}$$

The plot of parametrized surface patches uses the environment of `plot3d` described in part II, for the menu as well as for options.

Example of parametrized surface patch whose coordinates are given in the form of expressions:

```
> p:=u*cos(v)-u*sin(v)+u;q:=u*cos(v)+u*sin(v)-u;
  r:=u*sin(v)+u;
```

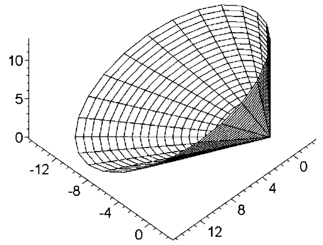
$$p := u\cos(v) - u\sin(v) + u$$

$$q := u\cos(v) + u\sin(v) - u$$

$$r := u\sin(v) + u$$

Ex.11

```
> plot3d([p,q,r], u=0..2*Pi, v=0..2*Pi,axes=framed,
  style=hidden);
```



Example of parametrized surface patch whose coordinates are given in the form of functions:

```
> f:=(u,v)->2*u*cos(v); g:=(u,v)->2*u*sin(v);
  h:=(u,v)->2*cos(2*v);
```

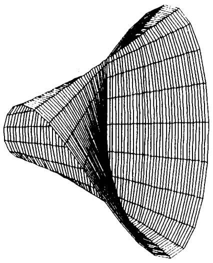
$$f := (u, v) \to 2\,u\cos(v)$$

$$g := (u, v) \to 2\,u\sin(v)$$

$$h := (u, v) \to 2\,\cos(2v)$$

Ex.12

```
> plot3d([f,g,h],0..1,0..2*Pi,
  grid=[35,35],scaling=constrained);
```



11.4 Surfaces Patches Parametrized in Cylindrical Coordinates

The function `plot3d` with the option `coords=cylindrical` can be used to represent a surface which is defined in cylindrical coordinates.

- Given an expression `p` depending on the two free variables `theta` and `z`, and the numerical values `a`, `b`, `c`, `d` such that `a<b` and `c<d`, the evaluation of `plot3d(p,theta=a..b,z=c..d,coords=cylindrical)` plots the surface which is defined in cylindrical coordinates by $r = p$ for θ ranging from a to b and z ranging from c to d, i.e. the surface which is defined parametrically by

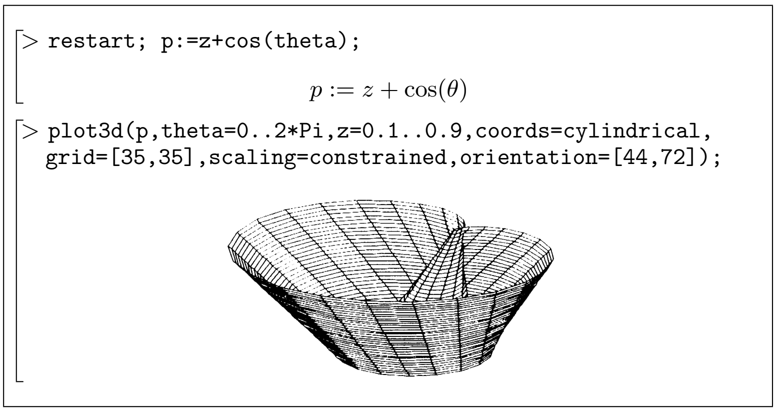
$$= \{(p\cos\theta, p\sin\theta, z) \mid a \quad \theta \quad b,\ c \quad z \quad d\}\,.$$

- Given expressions `p`, `alpha`, `q` depending on the free variables `u` and `v`, and the numerical values `a`, `b`, `c`, `d` such that `a<b` and `c<d`, `plot3d([p,alpha,q], u=a..b,v=c..d,coords=cylindrical)` plots the surface which is defined in cylindrical coordinates by $r = p$, $\theta =$ and $z = q$, for u ranging from a to b and v ranging from c to d, i.e. the surface which is defined parametrically by

$$= \{(p\ \cos\ ,\ p\ \sin\ ,\ q) \mid a \quad u \quad b,\ c \quad v \quad d\}\,.$$

Warning ! The option `coords=cylindrical` doesn't force the function `plot3d` to use an orthonormal frame. One must to specify `scaling=constrained` to obtain the plot with orthonormal axes.

Ex.13



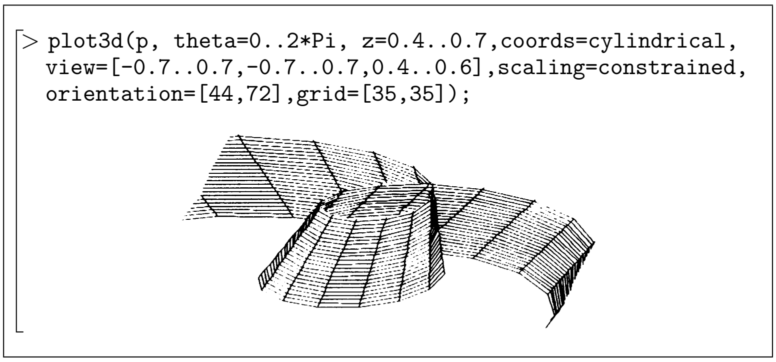
```
> restart; p:=z+cos(theta);
```

$$p := z + \cos(\theta)$$

```
> plot3d(p,theta=0..2*Pi,z=0.1..0.9,coords=cylindrical,
  grid=[35,35],scaling=constrained,orientation=[44,72]);
```

Plot of a part of the previous surface by using the option `view`:

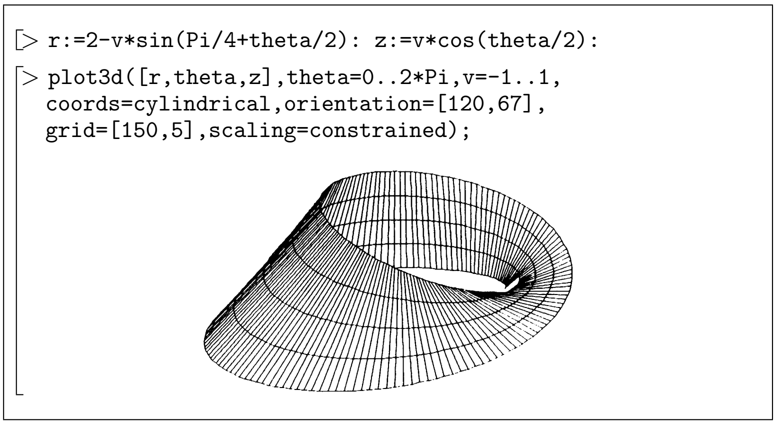
Ex.14



```
> plot3d(p, theta=0..2*Pi, z=0.4..0.7,coords=cylindrical,
  view=[-0.7..0.7,-0.7..0.7,0.4..0.6],scaling=constrained,
  orientation=[44,72],grid=[35,35]);
```

And to finish up, the classical Möbius strip:

Ex.15



```
> r:=2-v*sin(Pi/4+theta/2): z:=v*cos(theta/2):
> plot3d([r,theta,z],theta=0..2*Pi,v=-1..1,
  coords=cylindrical,orientation=[120,67],
  grid=[150,5],scaling=constrained);
```

11.5 Surface Patches Parametrized in Spherical Coordinates

The function `plot3d` with the option `coords=spherical` can be used to represent a surface which is defined in spherical coordinates.

- Given an expression `p` depending on two free variables `theta` and `phi` and numerical values `a`, `b`, `c` and `d` such that `a<b` and `c<d`, the evaluation of `plot3d(p,theta=a..b,phi=c..d,coords=spherical)` plots the surface which is defined in spherical coordinates by $r = p$ for θ ranging from a to b and ranging from c to d, i.e. the surface which is defined parametrically by

$$= \{(p \cos\theta \sin \ , p \sin\theta \sin \ , p \cos \ ) \mid a \quad \theta \quad b \text{ and } c \qquad d\}\,.$$

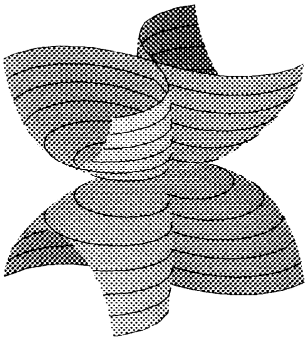
- Given expressions `p`, `alpha` and `beta` depending on the free variables `u` and `v` and numerical values `a`, `b`, `c` and `d` such that `a<b` and `c<d`, `plot3d([p,alpha,beta],u=a..b,v=c..d,coords=spherical)` plots the surface which is defined in spherical coordinates by $r = p$, $\theta =$ and $=$, for u ranging from a to b and v ranging from c to d, i.e. the surface which is defined parametrically by

$$= \{(p \cos \quad \sin \ , p \sin \quad \sin \ , p \cos \ ) \mid a \quad u \quad b \text{ and } c \quad v \quad d\}\,.$$

Warning ! The option `coords=spherical` doesn't force the function `plot3d` to use an orthonormal frame. One must specify `scaling=constrained` to obtain the plot with orthonormal axes.

Ex.16

```
> plot3d(cos(theta)*cos(phi),theta=0..2*Pi,phi=0..Pi,
  grid=[50,50],coords=spherical,orientation=[54,67],
  style=patchcontour,view=[-0.3..0.3,-0.5..0.5,-0.3..0.3],
  scaling=constrained);
```



11.6 Parametrized Space Curves

The function `spacecurve` is used to represent a parametrized space curve. It doesn't belong to the standard library and has therefore to be loaded with

Ex.17

```
> with(plots,spacecurve):            To load only the spacecurve
function
                                               of the plots library
```

11.6.1 Plot of a Parametrized Curve

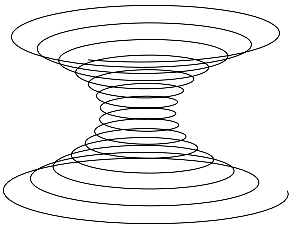
Given two numerical values `a` and `b` such that `a<b`,

- if `p`, `q` and `r` are expressions in a free variable `u` then the evaluation of `spacecurve([p,q,r,u=a..b])` plots the curve which is defined parametrically by $x = p$, $y = q$ and $z = r$ for u ranging from a to b.
- if `f`, `g` and `h` are functions in one variable, `spacecurve([f,g,h,a..b])` or `spacecurve([f(u),g(u),h(u),u=a..b])` plots the curve which is defined parametrically by $x = f(u)$, $y = g(u)$ and $z = \ (u)$ for u ranging from a to b.

The option `numpoints` can be used to increase the number of points MAPLE uses to plot the curve (by default `numpoints=50`). Except for the option `grid`, all options already described for the function `plot3d` can be used with `spacecurve`.

Ex.18

```
> with(plots,spacecurve):      To load the spacecurve function

> p:=(t^2+3)*sin(15*t):q:=(t^2+3)*cos(15*t):r:=t:
  spacecurve([p,q,r,t=-3..3],numpoints=800,
  orientation=[51,70]);
```



11.6.2 Simultaneous Plot of Several Parametrized Curves

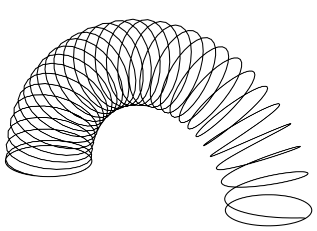
Given real numbers `a1`, `b1`, `a2` and `b2` such that `a1<b1` and `a2<b2` and expressions `p1`, `p2`, `q1`, `q2`, `r1` and `r2` in the free variable `u`, the evaluation of `spacecurve({[p1,q1,r1,u=a1..b1],[p2,q2,r2,u=a2..b2]})` plots the curve defined by $x = p1$, $y = q1$ and $z = r1$ for u ranging from $a1$ to $b1$ as well as the curve defined by $x = p2$, $y = q2$ and $z = r2$ for u ranging from $a2$ to $b2$.

This syntax can be generalized to any number of curves. Three curves are used in the following example.

Ex.19

```
> f:=sin(40*t^(4/3)):
  g:=(3+cos(40*t^(4/3)))*cos(t):
  h:=(3+cos(40*t^(4/3)))*sin(t):
```

```
> spacecurve({[f,g,h,t=0..Pi],
  [sin(t),3+cos(t),0,t=0..2*Pi],
  [sin(t),-3+cos(t),0,t=0..2*Pi]},scaling=constrained,
  orientation=[32,69],numpoints=800,shading=none);
```



11.7 Surfaces Defined Implicitly

The function `implicitplot3d` plots a surface which is defined implicitly. It doesn't belong to the standard library and must therefore be loaded with

Ex.20

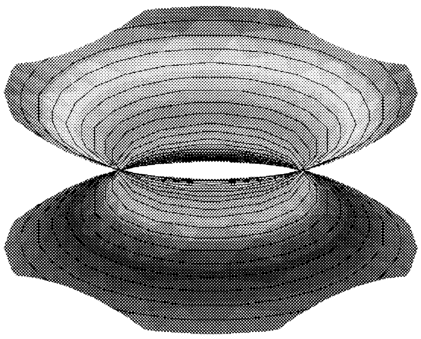
```
> restart; with(plots,implicitplot3d):
```

If `x1`, `x2`, `y1`, `y2`, `z1` and `z2` are six real numbers such that `x1<x2`, `y1<y2`, `z1<z2` and if `p` is an expression in the three free variables `x`, `y` and `z`, then `implicitplot3d(p,x=x1..x2,y=y1..y2,z=z1..z2)` plots the part of the surface defined by the equation $p = 0$ that is contained in $[x1, x2] \times [y1, y2]$ $\times [z1, z2]$.

The equation `p=0` and more generally the equation `p=q` can be used as first argument of `implicitplot3d`. All options already described for the function `plot3d` (p. 185) can be used for the function `implicitplot3d`.

Ex.21

```
> implicitplot3d(x^2+y^2-z^2-1,x=-2..2,y=-2..2,z=-2..2,
  style=patchcontour,shading=zhue);
```



The implicit equation of the surface to be plotted can also be given by a function. Given six real numbers x1, x2, y1, y2, z1, z2 such that x1<x2, y1<y2, z1<z2 and a function f in three variables, the evaluation of implicitplot3d(f,x1..x2,y1..y2,z1..z2) plots the part of the surface defined by the equation $f(x, y, z) = 0$ that is contained in the volume $[x1, x2] \times [y1, y2] \times [z1, z2]$.

The previous example could have been written

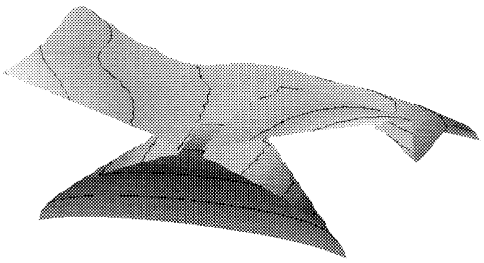
Ex.22

```
> f:=(x,y,z)->x^2+y^2-z^2-1;
> implicitplot3d(f,-2..2,-2..2,-2..2, style=patchcontour,
  shading=zhue);
```

As for implicit plane curves, grid can be used to improve the quality of plots of implicitly defined surfaces. The option grid=[m,n,p] defines a grid that has m points along x, n points along y and p points along z (grid=[10,10,10] by default). As for the 2D case, this option should be used with moderation!

Ex.23

```
> implicitplot3d(x^7+3*x^2*y^3+3*y^2*z^2+z^3,
  x=-0.5..0.5,y=-0.5..0.5,z=-0.5..0.5,grid=[20,20,20],
  orientation=[-72,49],style=patchcontour,shading=zhue);
```



11.8 Mixing Plots from Different Origins

As for the 2D case, the function display can be used to represent surfaces or curves created in different ways in the same picture: parametrized surface patches, surfaces defined implicitly ... The function display doesn't belong to the standard library and must therefore be loaded before its first use.

Before using the function display one must, as in the planar case, start by storing the different objects (surfaces, curves) in variables. The function display can then be called with the list of variables to be plotted as its first argument. The function display can be used with the options described on p. 185.

To use display with a sphere and a torus, we start by storing these surfaces into the variables S and T:

Ex.24

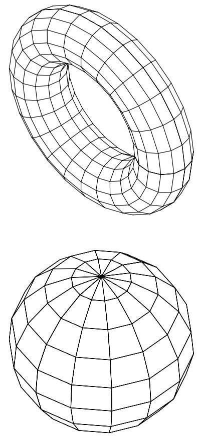
```
> f:=sin(u):g:=(3+cos(u))*cos(v)-3:h:=(3+cos(u))*sin(v):

> S:=plot3d(1.75,u=0..2*Pi,v=0..Pi,grid=[15,20],
  coords=spherical,scaling=constrained):

> T:=plot3d([f,g,h],u=0..2*Pi,v=0..2*Pi,grid=[15,25],
  scaling=constrained):
```

To verify the contents of S and T, we write:

Ex.25

```
> T; S;
```

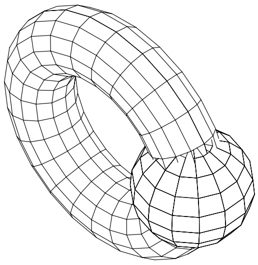
The evaluation of T and S returns the plots



With the option `style=hidden`, the function `display` returns the union of the surfaces without hidden parts

Ex.26

```
> with(plots,display):
> display([S,T],scaling=constrained,style=hidden);
```



We see the union of the two structures with the style `wireframe`

Ex.27

```
> display([S,T],scaling=constrained,style=wireframe);
```