Mini-Course: Computer Algebra for Physicists

Edgardo S. Cheb-Terrab
Maplesoft

This course is organized as a guided experience, 2 hours per day during five days, on learning the basics of the Maple language, and on using it to formulate algebraic computations we do in physics with paper and pencil. It is oriented to people not familiar with computer algebra (sections 1-5), as well as to people who are familiar but want to learn more about how to use it in Physics.

Motivation

Among other things, with computer algebra:

- You can concentrate more on the ideas (the model and its formulation) instead of on the algebraic manipulations
- You can extend your results with ease
- You can explore the mathematics surrounding your problem
- You can share your results in a reproducible way - and with that exchange about a problem in more productive ways
- After you learn the basics, the speed at which algebraic results are obtained with the computer compensates with dramatic advantage the extra time invested to formulate the problem in the computer.

All this *doesn't mean that we need* computer algebra, at all, but does mean computer algebra can enrich our working experience in significant ways.

What is computer algebra - how do you learn to use it?

Computer algebra is just another language to do the same computations you do with paper and pencil:

- There is our math/phys language, that we understand and use when doing computations with paper and pencil.

- There is a dictionary (the help pages) that translate the computer-algebra language into our math/phys language.

- We want to express - in the computer algebra language - those algebraic computations we do with paper and pencil using our math/phys language.

* For these purposes ***it is relevant that the computer algebra language is as close as possible to our math/phys language***.

* You ***only need to know a basic set of words and syntax***, and ***for everything else use the help pages*** (the dictionary). If the language is useful, ***with time you naturally remember more words***.

What is this mini-course about?

Below you will find 10 Sections with computational topics. Sections 1 to 5 are about the basics in computer algebra, a sort of minimum that we need to know. Sections 6 to 10 are dedicated to formulating, on the computer algebra worksheet, some of the typical algebraic computations in Physics.

Each section stats with a table summarizing it, followed by two subsections: one with examples (the introductory material I present in class, approx 15 minutes), and one with exercises including their solutions, the material you work with in classroom (approx 1 hour and 15 minuts). You can either try to solve the problems proposed by yourself or, frequently more convenient:

a Give a look at the solution presented - in that way refreshing the physics and getting ideas on a computational approach for the problem

b Close the solution subsection and try to solve the problem yourself (allow for repeating a and b as many times as you need to feel you can solve the problem thinking by yourself and understanding what you are doing).

In a first approach you can also only just give a look at all the solutions presented, to form a general idea. It is rewarding, however, to try to solve a minimum of one problem per section.

What can you expect from this mini-course?

- The number of topics is large; yet the idea is that you can get the minimum necessary to move ahead. The solved exercises help showing how you translate paper and pencil computations into computer algebra computations.
- In order to cover the topics proposed, some of the computational ideas and commands are introduced directly within the exercises, and some other ones are only indicated by pointing to help pages (remember: place the cursor over the command you don't know about, then press F2 to open a related help page). There you can selectively grab only the information you need, or explore furthermore if you prefer - this is very personal.
- At the end of these 5 lectures you will have seen the main ideas behind doing - with computer algebra - the typical algebraic computations of physics. You will also be able to either formulate a problem yourself, or to formulate a more precise question on what part of the computation you feel lost, and you will be able to make use of any good answer provided to your question.

Summary: Learning the whole language takes more than 5 days, but below you have all what you need to move by yourself + interact with others.

Explore. Having success doesn't matter, using your curiosity as a compass does - things can be done in so many different ways. Have full permission to fail. Share your insights. All questions are valid even if to the side. Computer algebra can transform the algebraic computation part of physics into interesting discoveries and fun.

1. Arithmetic operations and elementary functions

Operators:	+, -, *, /, ^
Functions	exp, ln, sin, cos, tan, csc, sec, cot, arcsin, arccos, arctan, arccsc, arcsec, arccot. For the hyperbolic functions put an h at the end as in sinh, arctanh, etc.
Manipulation commands	Related to numerical evaluation: evalf, Digits. The complex components: Re, Im, conjugate, abs, argument Related to functions: series, convert (any function to any other one when possible), FunctionAdvisor Related to plotting: plot, plot3d, plots:-plotcompare

Table 1: Arithmetic operations and elementary functions

Examples

Blank spaces mean multiplication.
Function application is represented with rounded parenthesis (), as in $f(x)$.
Indexation, as used in tensors, is represented with squared brackets [], as in A[mu] displayed as A_{mu}.
Numerical approximation is obtained applying *evalf*

```
> restart; interface(imaginaryunit = i) :
```

```
> 4 + 5 i
```

$$4 + 5\,i \qquad \textbf{(5.1.1)}$$

```
> Re((5.1.1))
```

$$4 \qquad \textbf{(5.1.2)}$$

```
> conjugate((5.1.1))
```

$$4 - 5\,i \qquad \textbf{(5.1.3)}$$

```
> evalf(Pi)
```

$$3.141592654 \qquad \textbf{(5.1.4)}$$

```
> Digits
```

$$10 \qquad \textbf{(5.1.5)}$$

```
> evalf[50](Pi)
```

$$3.1415926535897932384626433832795028841971693993751 \qquad \textbf{(5.1.6)}$$

```
> FunctionAdvisor( )
The usage is as follows:
    > FunctionAdvisor( topic, function, ... );
where 'topic' indicates the subject on which advice is
required, 'function' is the name of a Maple function, and '..
.' represents possible additional input depending on the
'topic' chosen. To list the possible topics:
    > FunctionAdvisor( topics );
A short form usage,
```

```
    > FunctionAdvisor( function );
with just the name of the function is also available and
displays a summary of information about the function.
```

```
> FunctionAdvisor(topic)
* Partial match of "topic" against topic "topics".
The topics on which information is available are:
```

[*DE*, *analytic_extension*, *asymptotic_expansion*, *branch_cuts*, *branch_points*, *calling_sequence*, *class_members*, *classify_function*, *definition*, *describe*, *differentiation_rule*, *function_classes*, *identities*, *integral_form*, *known_functions*, *relate*, *required_assumptions*, *series*, *singularities*, *special_values*, *specialize*, *sum_form*, *symmetries*, *synonyms*] **(5.1.7)**

```
> FunctionAdvisor(classes)
```

[*trig*, *trigh*, *arctrig*, *arctrigh*, *elementary*, *GAMMA_related*, *Psi_related*, *Kelvin*, *Airy*, *Hankel*, *Bessel_related*, *0F1*, *orthogonal_polynomials*, *Ei_related*, *erf_related*, *Kummer*, *Whittaker*, *Cylinder*, *1F1*, *Elliptic_related*, *Legendre*, *Chebyshev*, *2F1*, *Lommel*, *Struve_related*, *hypergeometric*, *Jacobi_related*, *InverseJacobi_related*, *Elliptic_doubly_periodic*, *Weierstrass_related*, *Zeta_related*, *Other*, *Bell*, *Heun*, *trigall*, *arctrigall*, *piecewise_related*, *complex_components*, *integral_transforms*] **(5.1.8)**

```
> FunctionAdvisor(ele)
* Partial match of "ele" against topic "elementary".
The 26 functions in the "elementary" class are:
```

[arccos, arccosh, arccot, arccoth, arccsc, arccsch, arcsec, arcsech, arcsin, arcsinh, arctan, arctanh, cos, cosh, cot, coth, csc, csch, exp, ln, sec, sech, sin, sinh, tan, tanh] **(5.1.9)**

```
> FunctionAdvisor(identities, sin)
```

$$\left[\sin(z) = -\sin(-z),\ \sin(z) = 2\sin\left(\frac{z}{2}\right)\cos\left(\frac{z}{2}\right),\ \sin(z) = \frac{1}{\csc(z)},\ \sin(z)\right. \qquad \textbf{(5.1.10)}$$

$$= \frac{2\tan\left(\frac{z}{2}\right)}{1+\tan\left(\frac{z}{2}\right)^2},\ \sin(z) = -\frac{i}{2}\left(e^{iz} - e^{-iz}\right),\ \sin(z)^2 = 1 - \cos(z)^2,\ \sin(z)^2$$

$$\left. = \frac{1}{2} - \frac{\cos(2z)}{2}\right]$$

```
> FunctionAdvisor(display, ln)
ln belongs to the class "elementary" and so, in principle, it
can be related to various of the 26 functions of that class -
see FunctionAdvisor( "elementary" );
```

$$describe = (\ln = natural\ logarithm)$$

$$classify\_function = elementary$$

$$definition = \left[\ln(z) = (z-1)\left(\sum_{\_k1=0}^{\infty} \frac{(1-z)^{-\_k1}}{1+\_k1}\right), \mathrm{And}(|z-1|<1)\right]$$

$$singularities = [\ln(z), No\ isolated\ singularities]$$

$$branch\_points = [\ln(z), z \in [0, \infty + \infty\, i]]$$

$$branch\_cuts = [\ln(z), z < 0]$$

$$special\_values = \left[\ln(-1) = i\,\pi, \ln(1) = 0, \ln(-e) = 1 + i\,\pi, \ln(i) = \frac{i}{2}\pi, \ln(-i) = -\frac{i}{2}\pi, \ln(\infty) = \infty, \ln(-\infty) = \infty + i\,\pi, \left[\ln\left(e^{n}\right) = n, n::real\right]\right]$$

$$identities = [\ln(z) = i\,arg(z) + \ln(|z|)]$$

$$integral\_form = \left[\ln(z) = \int_1^z \frac{1}{\_k1}\, d\_k1, \mathrm{And}(\mathbf{Not}\, z < 0)\right]$$

$$differentiation\_rule = \left(\frac{d}{dz}\ln(f(z)) = \frac{\frac{d}{dz}f(z)}{f(z)}\right)$$

$$DE = \left[f(z) = \ln(z), \left[\frac{d}{dz}f(z) = \frac{1}{z}\right]\right]$$

$$series = (series(\ln(z), z, 4) = \ln(z))$$

$$asymptotic\_expansion = (asympt(\ln(z), z, 4) = \ln(z)) \quad \textbf{(5.1.11)}$$

> $\cos(x) + i\sin(x)$

$$\cos(x) + i\sin(x) \quad \textbf{(5.1.12)}$$

> $convert(\%, \exp)$

$$e^{i\,x} \quad \textbf{(5.1.13)}$$

> $convert(\%, trig)$

$$\cos(x) + i\sin(x) \quad \textbf{(5.1.14)}$$

> $FunctionAdvisor(relate, \arcsin, \ln)$

$$\arcsin(z) = -i\ln\left(\sqrt{-z^2+1} + i\,z\right) \quad \textbf{(5.1.15)}$$

> $FunctionAdvisor(specialize, \arcsin)$

$$\left[\arcsin(z) = \frac{z\,HC\left(0, \frac{1}{2}, 0, 0, \frac{1}{4}, \frac{z^2}{z^2-1}\right)}{\sqrt{-z^2+1}}, with\ no\ restrictions\ on\ (z)\right], \left[\arcsin(z)\right. \quad \textbf{(5.1.16)}$$

$$\left. = z\,HG\left(0, 0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}, z^2\right), with\ no\ restrictions\ on\ (z)\right], \left[\arcsin(z) = \frac{\pi}{2}\right.$$

$$+\frac{am^{-1}(\operatorname{arcsec}(z)|1)\ (z-1)}{\sqrt{-(z-1)^2}},\ \operatorname{And}(\Re(z)\in(0,\pi))\Bigg],\Bigg[\arcsin(z)$$

$$=\frac{z\,\pi\,P^{\left(\frac{1}{2},-\frac{1}{2}\right)}_{-\frac{1}{2}}\left(-2z^2+1\right)}{2},\ \textit{with no restrictions on}\ (z)\Bigg],\Bigg[\arcsin(z)$$

$$=\frac{z\sqrt{\pi}\ \left(-2z^2+2\right)^{1/4}P^{-\frac{1}{2}}_{-\frac{1}{2}}\left(-2z^2+1\right)}{2\left(-2z^2\right)^{1/4}},\ \textit{with no restrictions on}\ (z)\Bigg],$$

$$\left[\arcsin(z)=\frac{z\,G^{1,2}_{2,2}\left(-z^2\middle|\begin{matrix}\frac{1}{2},\frac{1}{2}\\0,-\frac{1}{2}\end{matrix}\right)}{2\sqrt{\pi}},\ \textit{with no restrictions on}\ (z)\right],\left[\arcsin(z)=\frac{\pi}{2}\right.$$

$$-\arccos(z),\ \textit{with no restrictions on}\ (z)\Bigg],\Bigg[\arcsin(z)=\frac{\pi}{2}$$

$$+\frac{\operatorname{arccosh}(z)\ (z-1)}{\sqrt{-(z-1)^2}},\ \textit{with no restrictions on}\ (z)\Bigg],\Bigg[\arcsin(z)=\pi$$

$$-2\operatorname{arccot}\left(\frac{z}{1+\sqrt{-z^2+1}}\right),\ \textit{with no restrictions on}\ (z)\Bigg],\Bigg[\arcsin(z)$$

$$=\frac{1}{\mathrm{i}z+\sqrt{-z^2+1}+1}\left(\left(2\,\mathrm{i}\sqrt{-z^2+1}+2\,\mathrm{i}\right.\right.$$

$$\left.-2z\right)\operatorname{arccoth}\left(\frac{-\mathrm{i}z}{1+\sqrt{-z^2+1}}\right)+\mathrm{i}\pi\left(1\right.$$

$$\left.\left.+\sqrt{-z^2+1}\right)\sqrt{-\left(\frac{\mathrm{i}z}{1+\sqrt{-z^2+1}}+1\right)^2}\right),\ \textit{with no restrictions on}\ (z)\Bigg],$$

$$\left[\arcsin(z)=\operatorname{arccsc}\left(\frac{1}{z}\right),\ \textit{with no restrictions on}\ (z)\right],\left[\arcsin(z)\right.$$

$$\left.=\mathrm{i}\operatorname{arccsch}\left(\frac{\mathrm{i}}{z}\right),\ \textit{with no restrictions on}\ (z)\right],\left[\arcsin(z)=\frac{\pi}{2}-\operatorname{arcsec}\left(\frac{1}{z}\right),\right.$$

$$\left[\textit{with no restrictions on } (z)\right], \left[\arcsin(z) = \frac{\pi}{2} + \frac{\operatorname{arcsech}\left(\frac{1}{z}\right)(-1+z)}{\sqrt{-\left(\frac{1}{z}-1\right)^2 z^2}}\right],$$

$$\left[\textit{with no restrictions on } (z)\right], [\arcsin(z) = -\mathrm{i}\,\operatorname{arcsinh}(\mathrm{i}\,z),$$

$$\left[\textit{with no restrictions on } (z)\right], \left[\arcsin(z) = 2\arctan\left(\frac{z}{1+\sqrt{-z^2+1}}\right)\right],$$

$$\left[\textit{with no restrictions on } (z)\right], \left[\arcsin(z) = -2\,\mathrm{i}\,\operatorname{arctanh}\left(\frac{\mathrm{i}\,z}{1+\sqrt{-z^2+1}}\right)\right],$$

$$\left[\textit{with no restrictions on } (z)\right], \left[\arcsin(z) = z\,{}_2F_1\left(\frac{1}{2}, \frac{1}{2}; \frac{3}{2}; z^2\right)\right],$$

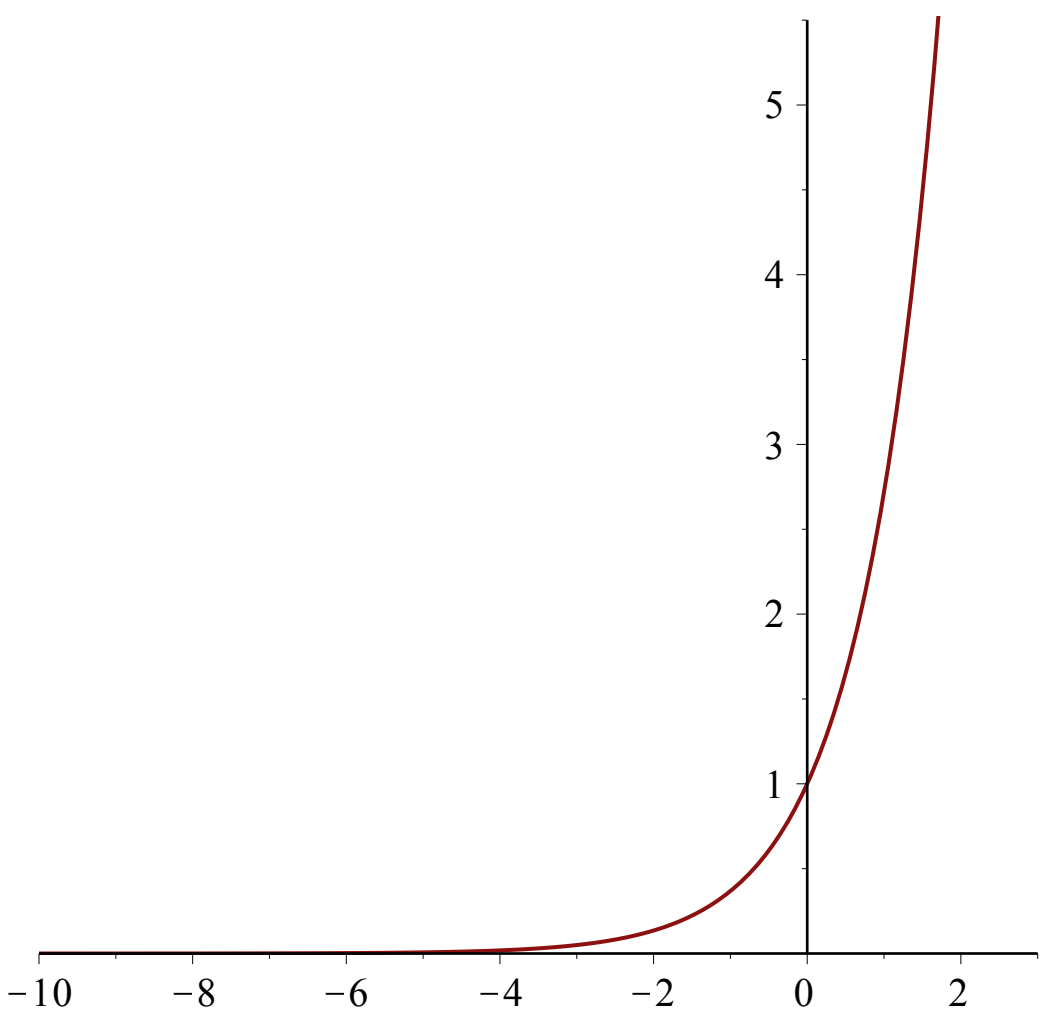
$$\left[\textit{with no restrictions on } (z)\right], \left[\arcsin(z) = -\mathrm{i}\ln\left(\sqrt{-z^2+1} + \mathrm{i}\,z\right)\right],$$

$$\left[\textit{with no restrictions on } (z)\right]$$

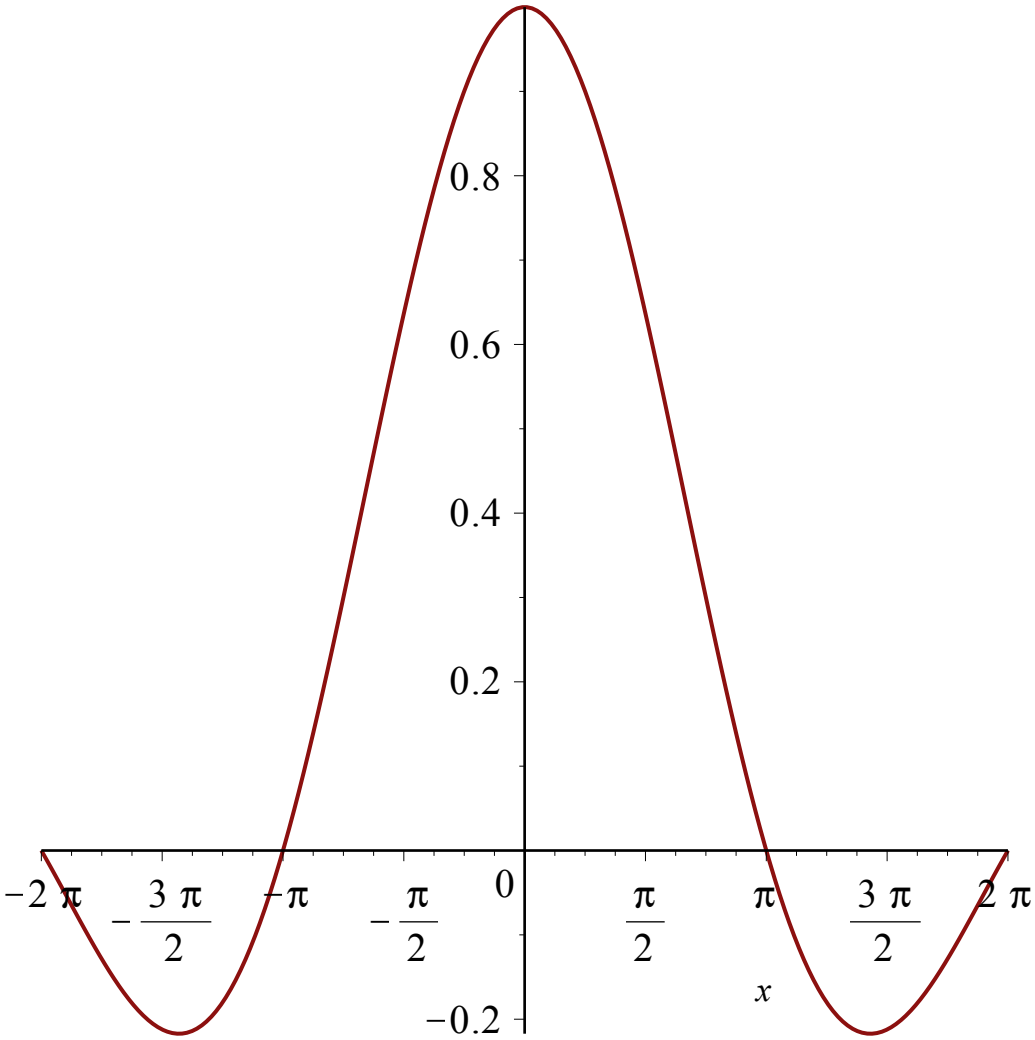
Plotting

2D plotting

```
> plot(exp)
```

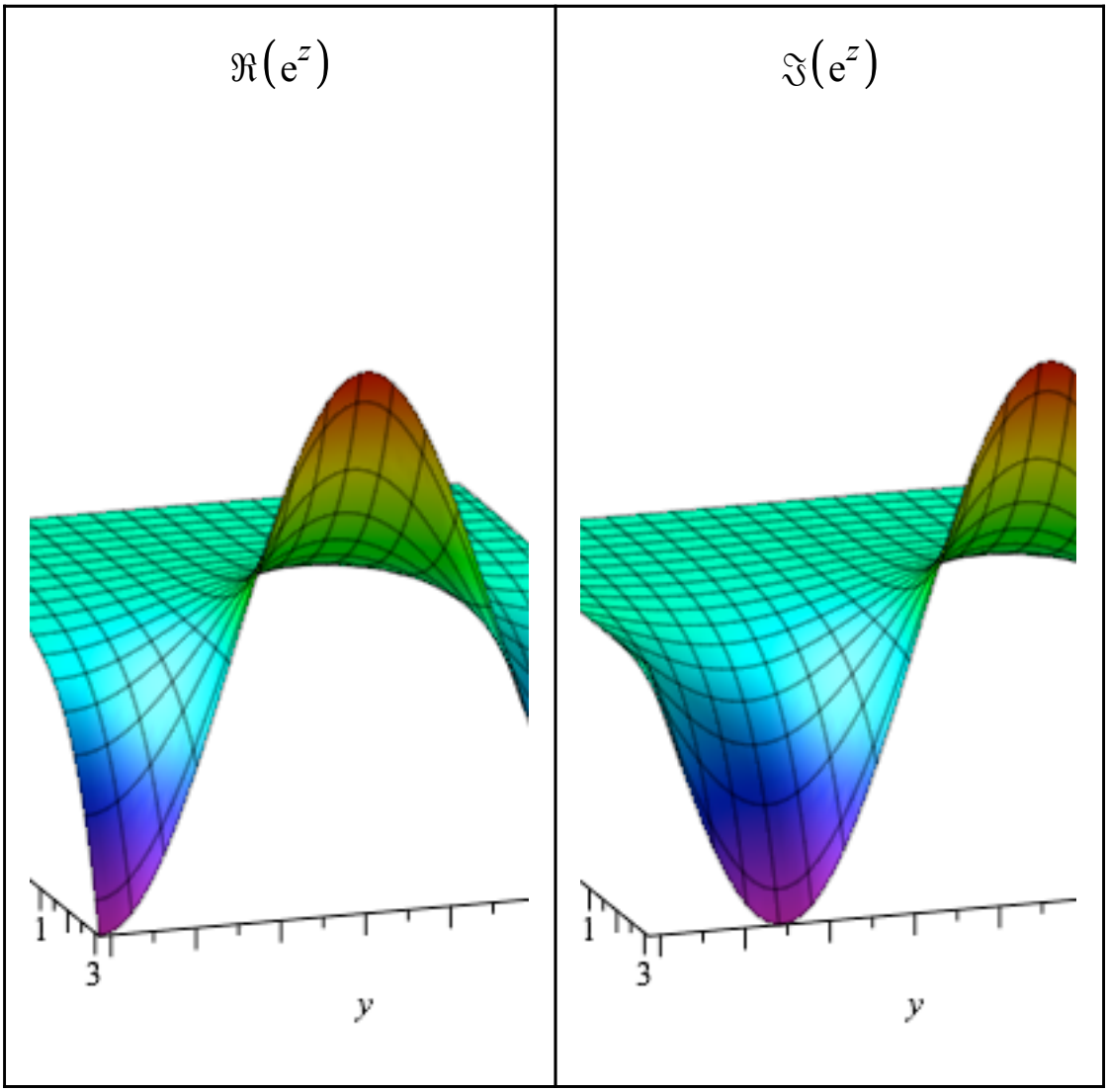


> $plot\left(\frac{\sin(x)}{x}\right)$



Complex algebraic expressions $F(z)$ of a complex variable z can be represented by two 3D plots: the value of $\Re(F(z))$ and $\Im(F(z))$ (so the real or imaginary parts of the expression) on the vertical axis and the real and imaginary parts of the variable z on the two horizontal axes (PlotExpression uses plots[plotcompare] with the option *expression_plot*)

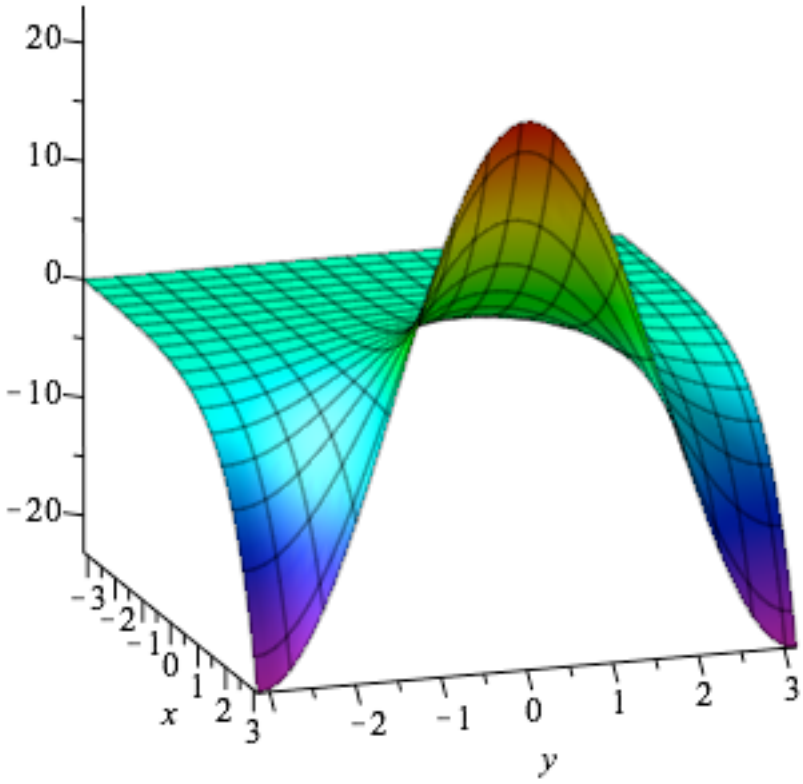
```
> PlotExpression := f → plots:-plotcompare(f, 0, _rest, 'expression_plot', 5) :
> PlotExpression(exp(z), scale_range = Pi, 5)
```



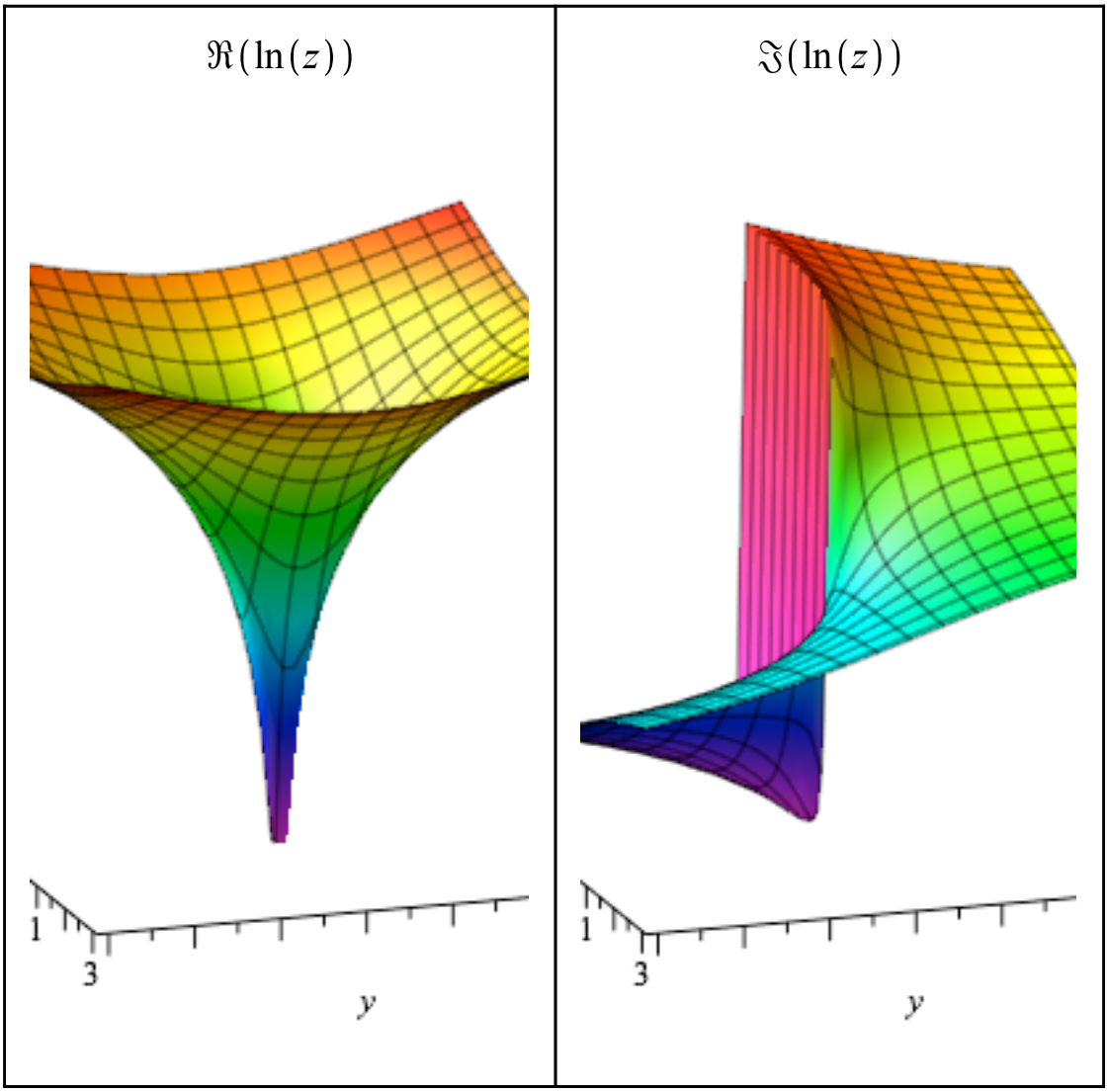
You can manipulate each of these plots 'in-situ', or reproduce them for independent manipulation:

```
> _P[1]
```

$$\Re\left(e^{z}\right)$$



> *PlotExpression*(ln(*z*), *scale_range* = Pi, 5)



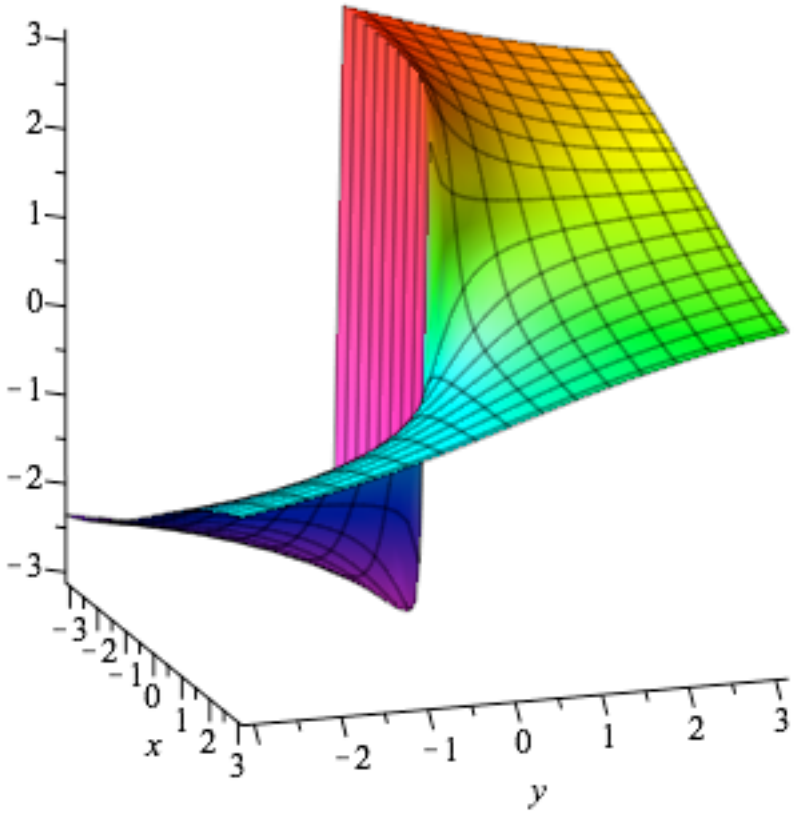
So ln is a multivalued function with a cut over

> *FunctionAdvisor*(*branch_cuts*, ln)

$$[\ln(z), z < 0] \qquad \textbf{(5.1.1.1)}$$

> _*P*[2]

$$\Im(\ln(z))$$



> $\ln(-I\cdot 10.0^{(-6)})$

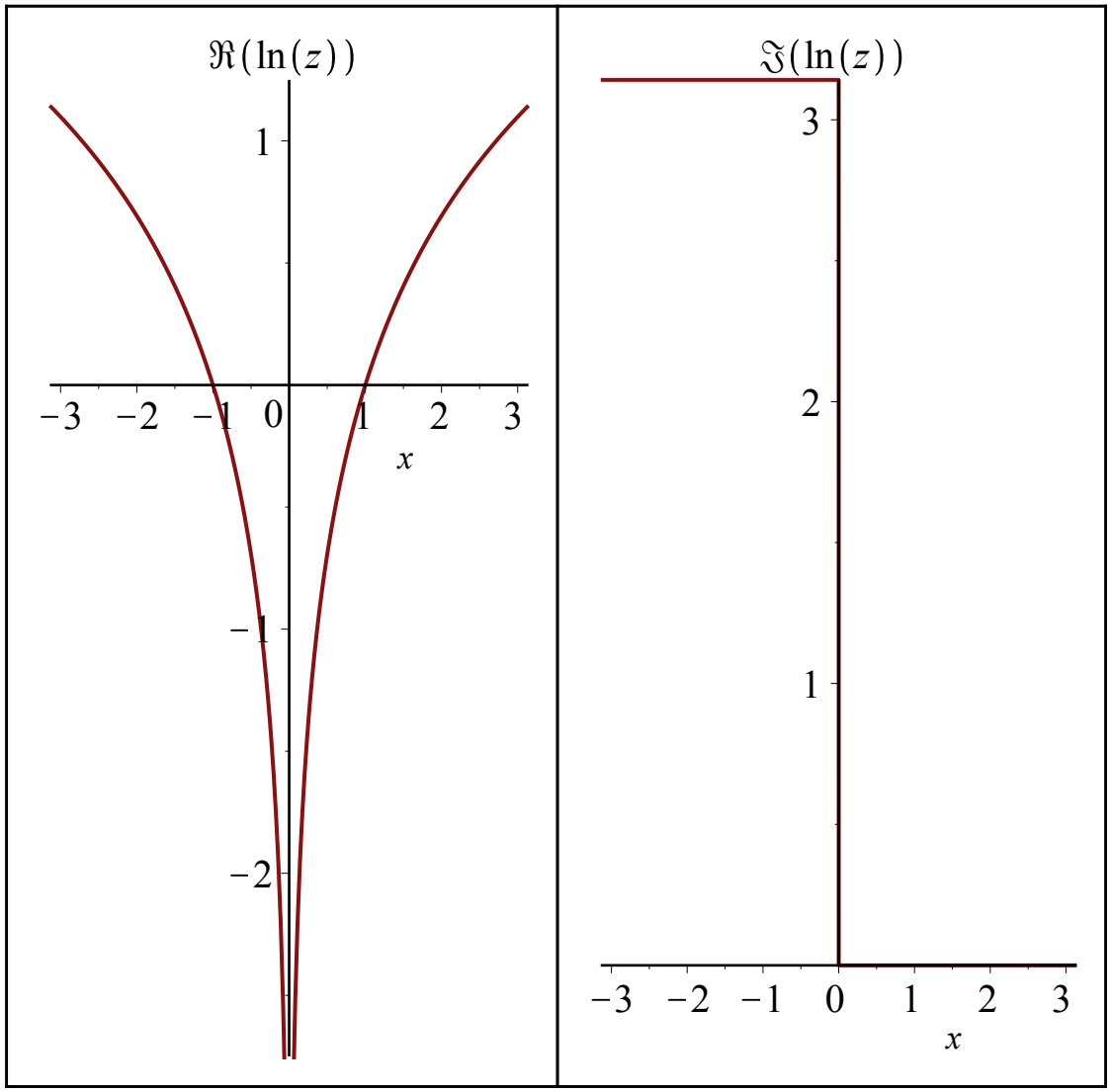
$$\ln(-1.000000000\ 10^{-6}\ I) \qquad \textbf{(5.1.1.2)}$$

> $\ln(+I\cdot 10.0^{(-6)})$

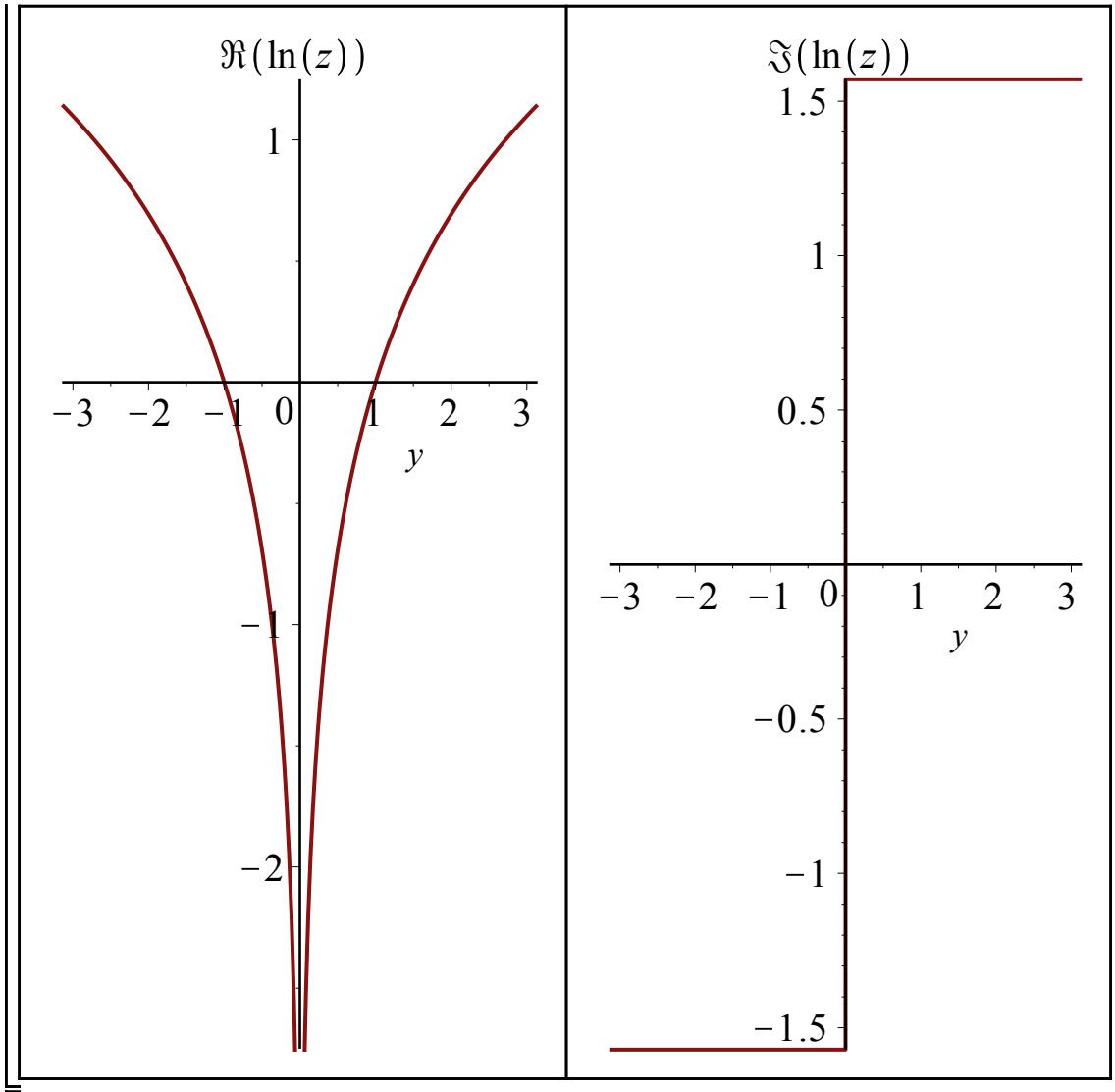
$$\ln(1.000000000\ 10^{-6}\ I) \qquad \textbf{(5.1.1.3)}$$

If you know in advance that z is real, or imaginary then the two 3D plots transform into two 2D plots:

> *PlotExpression*$(\ln(z), scale\_range = \text{Pi}, 5)$ assuming $z :: real$



> *PlotExpression*(ln(*z*), *scale_range* = Pi, 5) assuming *z* :: *imaginary*



>

Exercises

Choose one exercise, try to solve it in up to 10 minutes. If there is time, move to next problem. Or, feel free to use the time to explore the help pages about any related topic more of your interest.

1 Use the convert command to express the functions of the following groups in terms of each other

a [exp, sin, cos, tan, sec, csc, cot]

b [ln, arcsin, arccos, arctan arcsec, arccsc, arcot]

c Choose a couple of the relations between functions say A = B obtained and verify that A and B have the same series expansion

Solution

Remember always to restart to avoid other computations to interfere with the one you are developing

```
> restart
> convert(exp(z), sin)
```

$$\sin\left(I z + \frac{\pi}{2}\right) - I \sin(I z) \qquad \textbf{(5.2.1.1)}$$

```
> convert(exp(z), trig)
```

$$\cosh(z) + \sinh(z) \qquad \textbf{(5.2.1.2)}$$

```
> convert(exp(I·z), trig)
```

$$\cos(z) + I \sin(z) \qquad \textbf{(5.2.1.3)}$$

```
> convert(sin(z), exp)
```

$$-\frac{I}{2}\left(e^{I z} - e^{-I z}\right) \qquad \textbf{(5.2.1.4)}$$

```
> exp(ln(z))
```

$$z \qquad \textbf{(5.2.1.5)}$$

```
> sin(arcsin(z))
```

$$z \qquad \textbf{(5.2.1.6)}$$

```
> convert(ln(z), arcsin)
```

$$\frac{\sqrt{-z^2}\,\sqrt{\frac{z-1}{z+1}}\,\sqrt{\frac{z+1}{z-1}}\left(\arcsin\left(\frac{z^2+1}{2z}\right) - \frac{\pi}{2}\right)}{z} + I\pi\left(1 - \frac{\sqrt{-1-z}\,\sqrt{z}}{\sqrt{-z}\,\sqrt{z+1}}\right) \qquad \textbf{(5.2.1.7)}$$

```
> convert(arcsin(z), ln)
```

$$-I \ln\left(\sqrt{-z^2+1} + I z\right) \qquad \textbf{(5.2.1.8)}$$

```
> series(arcsin(z), z)
```

$$z + \frac{1}{6} z^3 + \frac{3}{40} z^5 + O\left(z^7\right) \qquad \textbf{(5.2.1.9)}$$

```
> series((5.2.1.8), z)
```

$$z + \frac{1}{6} z^3 + \frac{3}{40} z^5 + O\left(z^7\right) \qquad \textbf{(5.2.1.10)}$$

```
>
```

2 *plot* the *sin* function between $-$ pi and pi, then:

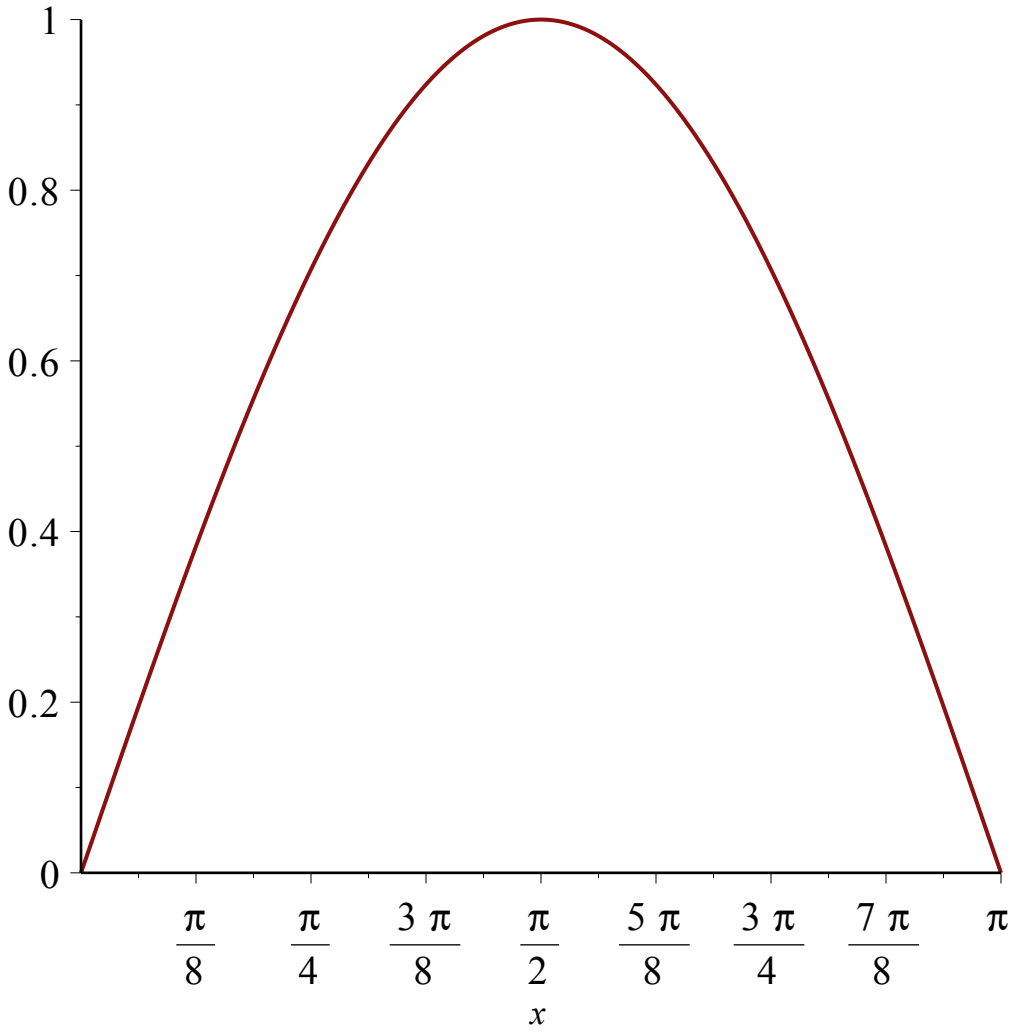
1. click the *plot* to select it ;
2. go to the menu Plot -> Probe Info -> Cursor position to access the probe tool;
3. use the probe to identify the coordinates of points of the *plot* between , as in $[[x_1, y_1], [x_2, y_2], `...`, [x_n, y_n]]$.
4. search the *help* system for '*interpolate*' and choose a command to interpolate a polynomial approximating $\sin(x)$ between $-$ pi and pi . How many points do you need to obtain an approximation more or less acceptable?

Solution

```
> restart
```

Because sin has a symmetry, you only need to investigate the problem in one semiplane - say between 0 and pi

```
> plot(sin(x), x = 0 ..Pi)
```



Again, because this plot also has a symmetry you only need to probe a few points to the left of $\frac{pi}{2}$ and because we know the value of the function at 0 and $\frac{pi}{2}$ and we are lazy we only choose probing at two points in between, so at $\frac{pi}{8}$ and $\frac{pi}{4}$; from there we get the values at $\frac{3\ pi}{4}, \frac{7\ pi}{8}$

```
> R := [[0, 0], [Pi/8, 0.368], [Pi/4, 0.7], [Pi/2, 1], [3*Pi/4, 0.7], [7*Pi/8, 0.368], [Pi, 0]]
```

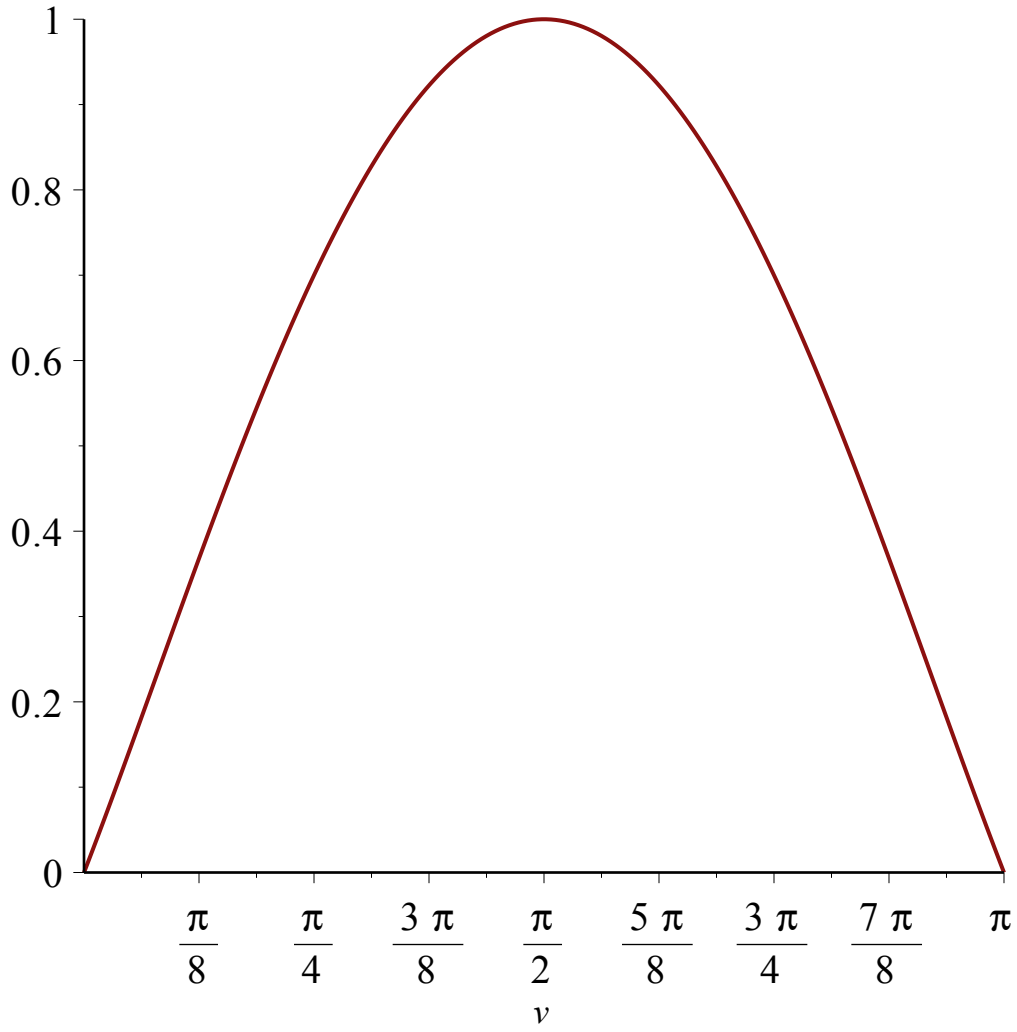
$$R := \left[[0, 0], \left[\frac{\pi}{8}, 0.368\right], \left[\frac{\pi}{4}, 0.7\right], \left[\frac{\pi}{2}, 1\right], \left[\frac{3\,\pi}{4}, 0.7\right], \left[\frac{7\,\pi}{8}, 0.368\right], [\pi, 0]\right] \quad \textbf{(5.2.2.1)}$$

```
> CurveFitting:-PolynomialInterpolation(R, v)
```

(5.2.2.2)

$$\frac{3.224787139\,v^6}{\pi^6} - \frac{9.674361452\,v^5}{\pi^5} + \frac{15.35187256\,v^4}{\pi^4} - \frac{14.57980932\,v^3}{\pi^3} + \frac{2.895339645\,v^2}{\pi^2} + \frac{2.782171431\,v}{\pi} \quad \textbf{(5.2.2.2)}$$

> $plot(\textbf{(5.2.2.2)}, v = 0..\mathrm{Pi})$



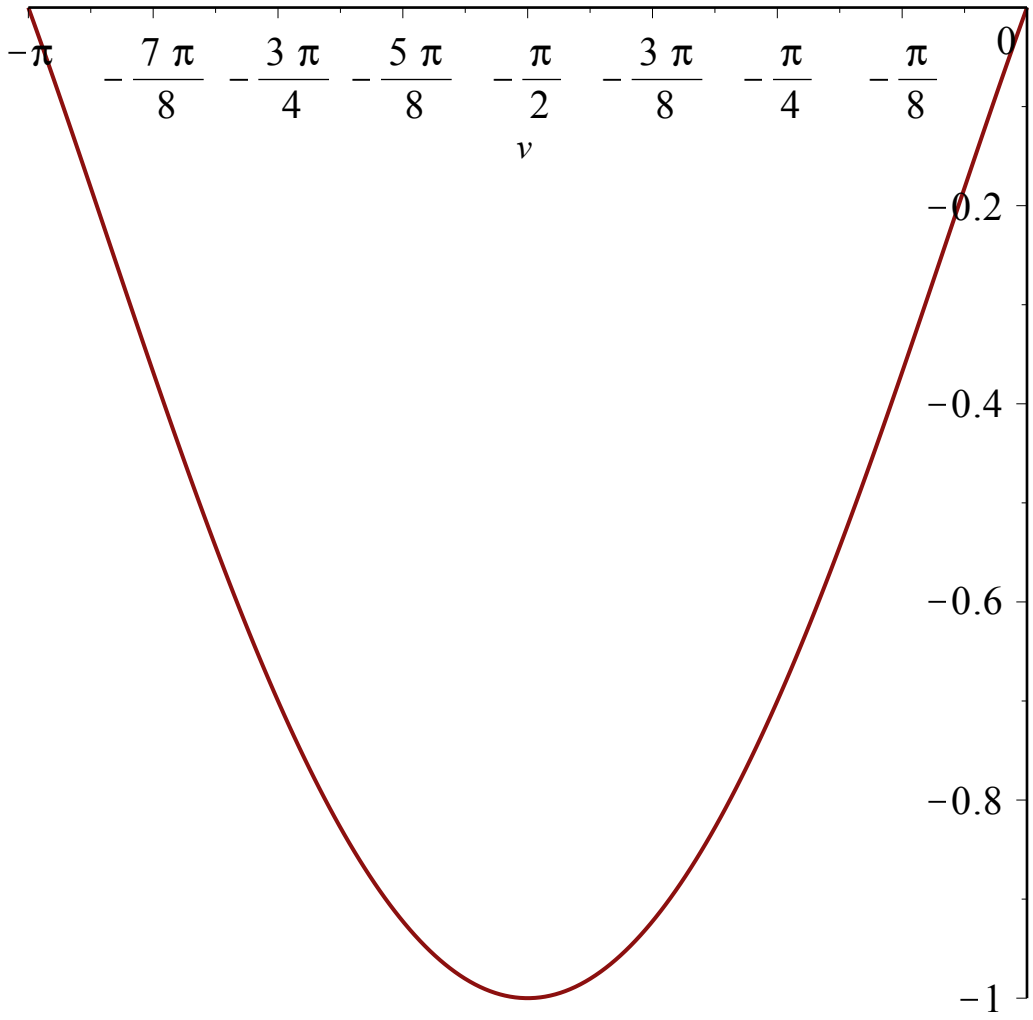
From the symmetry of sin, the interpolation from $-$ pi to 0 is

$$-\pi \quad \textbf{(5.2.2.3)}$$

> $-subs(v = -v, \textbf{(5.2.2.2)})$

$$-\frac{3.224787139\,v^6}{\pi^6} - \frac{9.674361452\,v^5}{\pi^5} - \frac{15.35187256\,v^4}{\pi^4} - \frac{14.57980932\,v^3}{\pi^3} - \frac{2.895339645\,v^2}{\pi^2} + \frac{2.782171431\,v}{\pi} \quad \textbf{(5.2.2.4)}$$

> $plot(\textbf{(5.2.2.4)}, v = -\mathrm{Pi}..0)$

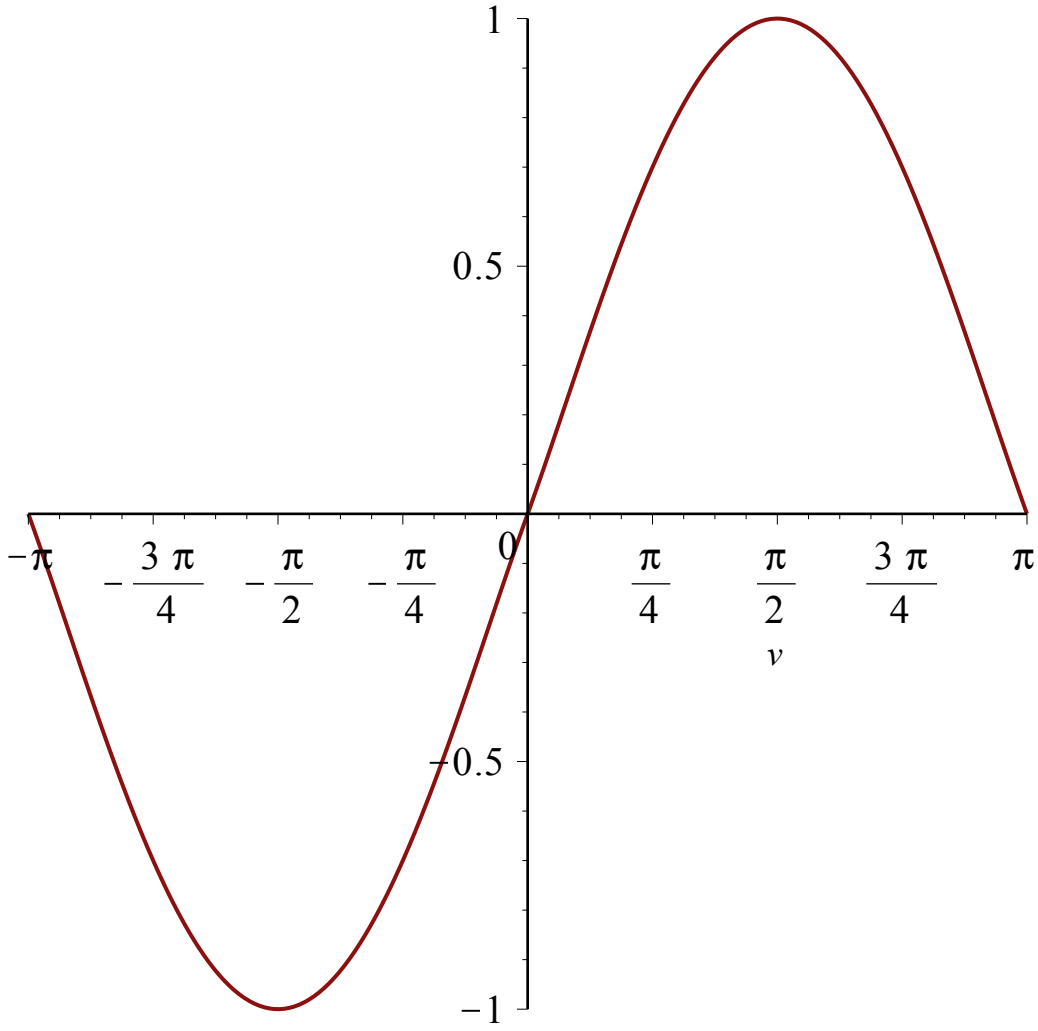


You can now construct the plot between -Pi to Pi putting the two interpolations into a piecewise function

> $piecewise(v \geq 0, \mathbf{(5.2.2.2)}, \mathbf{(5.2.2.4)})$

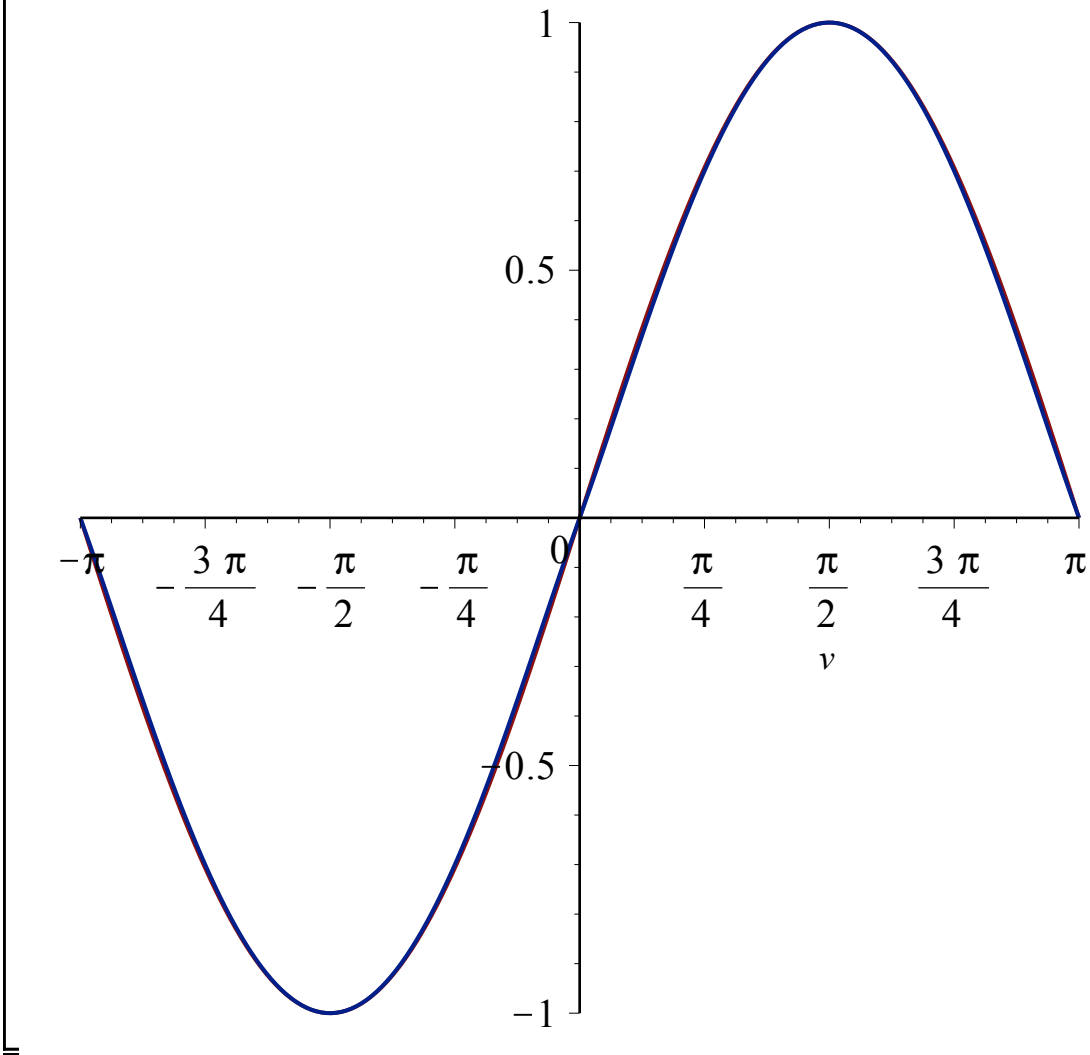
$$\begin{cases} \frac{3.224787139\, v^6}{\pi^6} - \frac{9.674361452\, v^5}{\pi^5} + \frac{15.35187256\, v^4}{\pi^4} - \frac{14.57980932\, v^3}{\pi^3} + \frac{2.895339645\, v^2}{\pi^2} \\ -\frac{3.224787139\, v^6}{\pi^6} - \frac{9.674361452\, v^5}{\pi^5} - \frac{15.35187256\, v^4}{\pi^4} - \frac{14.57980932\, v^3}{\pi^3} - \frac{2.895339645\, v^2}{\pi^2} \end{cases}$$

> $plot(\mathbf{(5.2.2.5)}, v = -\mathrm{Pi}..\mathrm{Pi})$



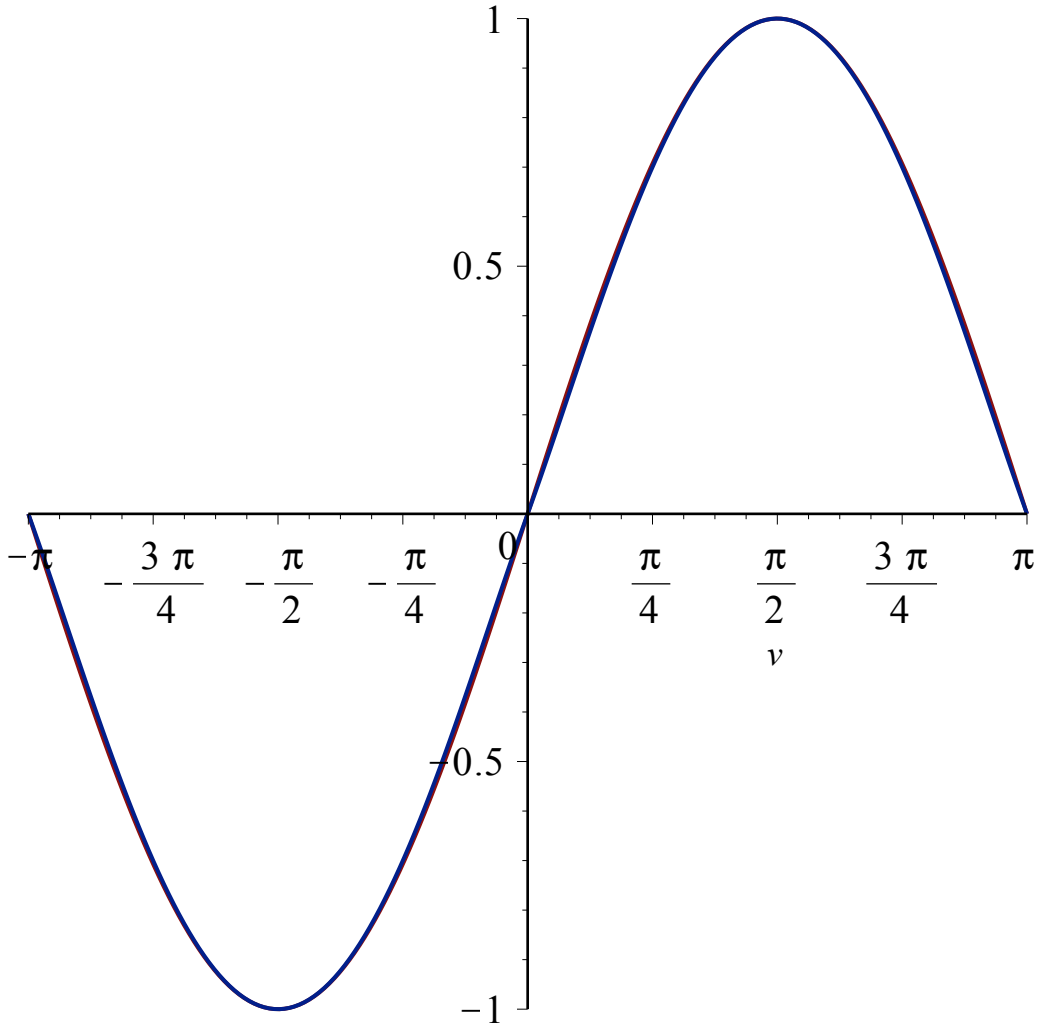
A plot superimposing sin(v) with its interpolation

```
> plot( [sin(v), (5.2.2.5)], v=-Pi..Pi)
```



You can now click the plot to display to plotting toolbar, and use the scale and translate plot tools to distinguish between these two lines. It is sometimes convenient to duplicate the plot to keep the complete picture

> *plot*([sin(*v*), **(5.2.2.5)**], *v* = -Pi..Pi)



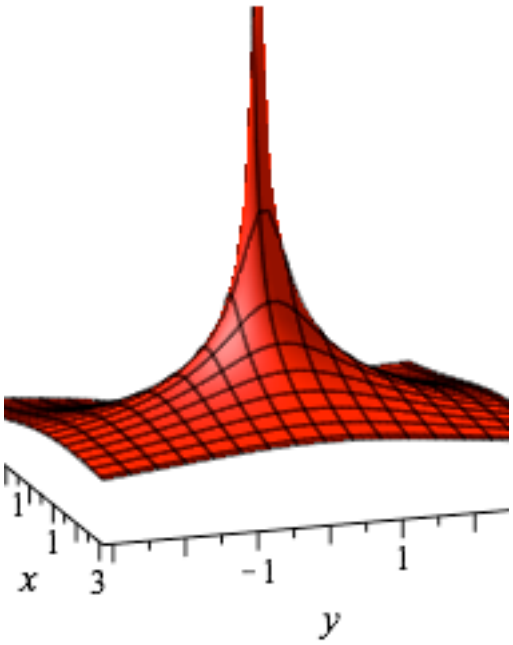
>

3 Use *plots:-plotcompare* to determine for which values of z you have $\frac{1}{\sqrt{z}} \neq \sqrt{\frac{1}{z}}$. Try it with the options same_box, *assuming* z::real and *assuming* z::imaginary

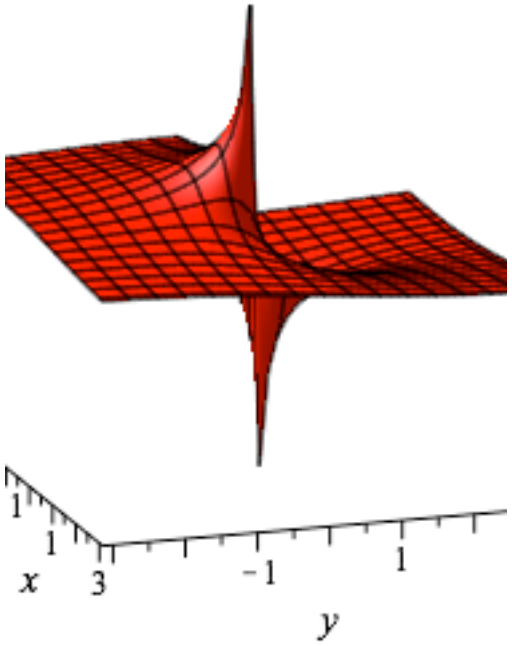
Solution

```
> restart
> plots:-plotcompare(1/sqrt(z), sqrt(1/z), same_box, scale_range = Pi)
```

$\Re\left(\frac{1}{\sqrt{x+\mathrm{I}\,y}}\right)$ **and** $\Re\left(\sqrt{\frac{1}{x+\mathrm{I}\,y}}\right)$

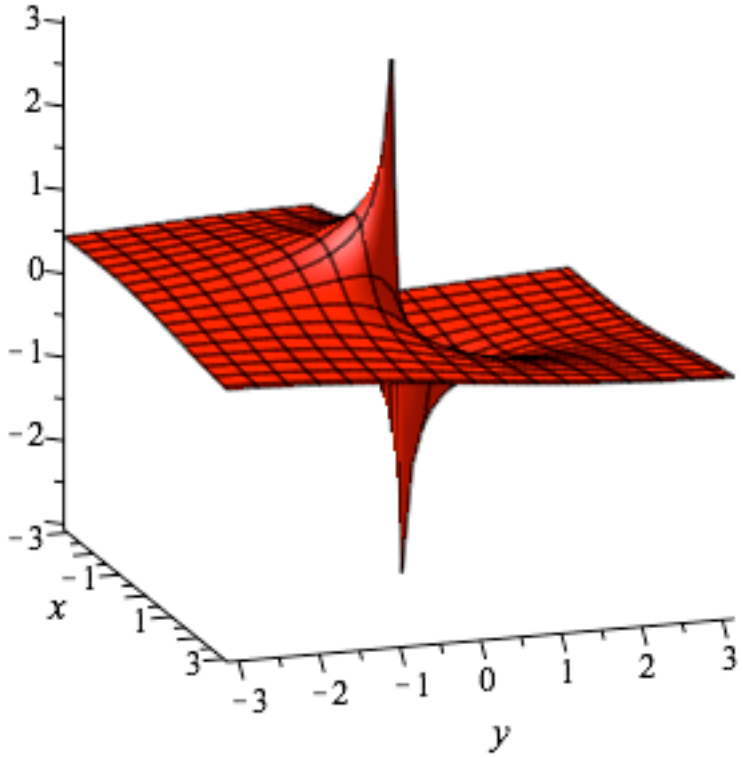


$\Im\left(\frac{1}{\sqrt{x+\mathrm{I}\,y}}\right)$ **and** $\Im\left(\sqrt{\frac{1}{x+\mathrm{I}\,y}}\right)$



> _P[2]

$$\Im\left(\frac{1}{\sqrt{x+\mathrm{I}\,y}}\right) \textbf{ and } \Im\left(\sqrt{\frac{1}{x+\mathrm{I}\,y}}\right)$$



So, apparently, these two expressions, $\frac{1}{\mathrm{sqrt}(z)}$, $\mathrm{sqrt}\left(\frac{1}{z}\right)$, are equal, BUT: there is a branch cut .... you need to check the values over the cut, that is located in the same place for the two expressions:

> $FunctionAdvisor\left(branch\_cuts, \mathrm{sqrt}\left(\frac{1}{z}\right)\right)$

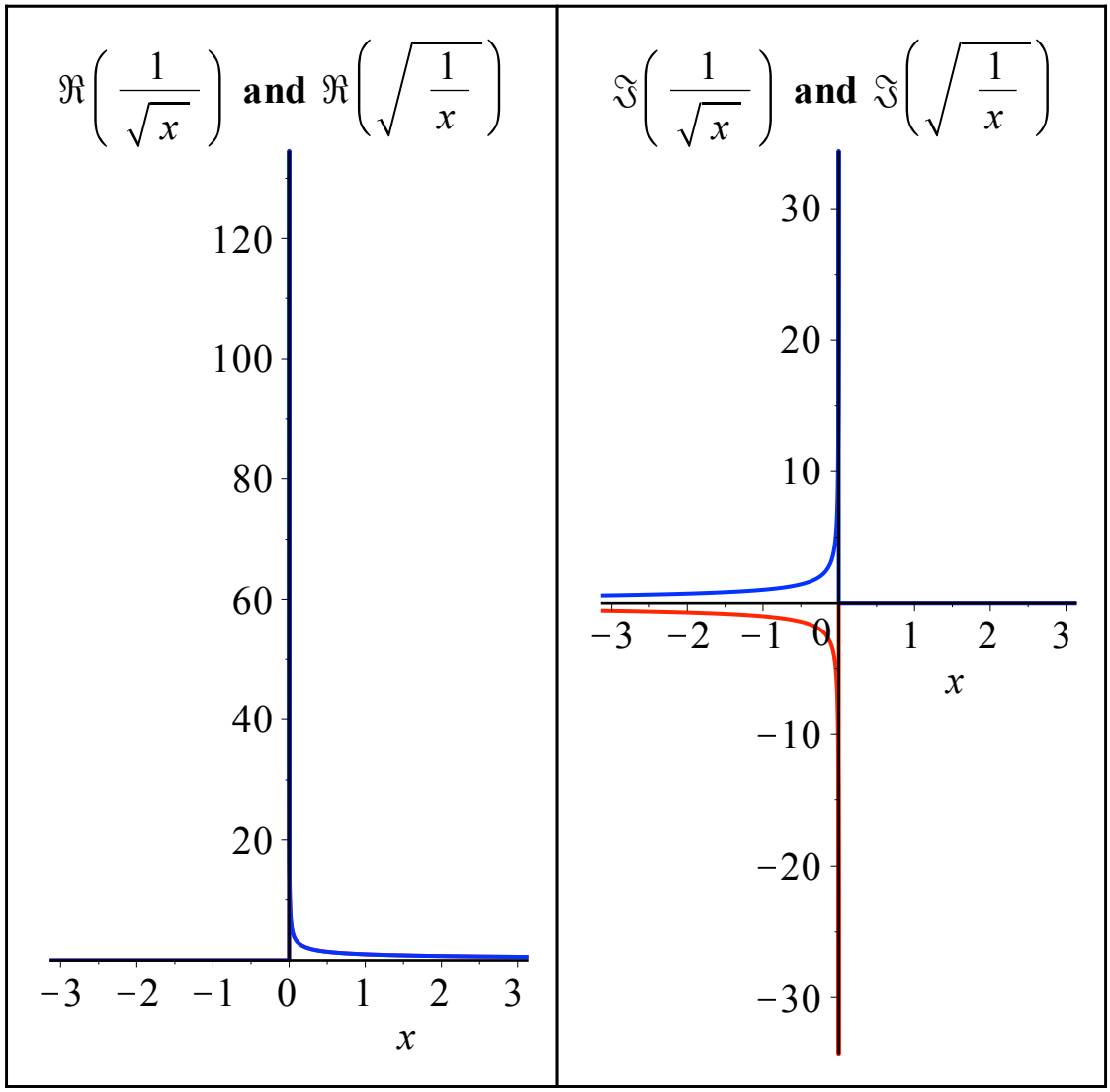
$$\left[\sqrt{\frac{1}{z}}, z < 0\right] \qquad \textbf{(5.2.3.1)}$$

> $FunctionAdvisor\left(branch\_cuts, \frac{1}{\mathrm{sqrt}(z)}\right)$

$$\left[\frac{1}{\sqrt{z}}, z < 0\right] \qquad \textbf{(5.2.3.2)}$$

For cuts over the real axis, the solution is to plot assuming real

> $plots\text{:-}plotcompare\left(\frac{1}{\mathrm{sqrt}(z)}, \mathrm{sqrt}\left(\frac{1}{z}\right), same\_box, scale\_range = \mathrm{Pi}\right)$ assuming z :: $real$



So: over the negative real axis, the imaginary parts of these two expressions have the same absolute value but different sign

```
> eval(1/sqrt(x), x = -0.3)
```

-1.825741858 I **(5.2.3.3)**

```
> eval(sqrt(1/x), x = -0.3)
```

1.825741858 I **(5.2.3.4)**

```
>
```

2. Algebraic Expressions, Equations and Functions

Algebraic expression	any mathematical object built with numbers, symbols and functions combined using arithmetic operations

Equation	a construction using the = sign, typically with algebraic expressions on the left-hand and right-hand sides
Function	It can be a known function as $\ln(z)$ or $J_n(z)$, or an unknown function, for example $f(x, y, z, t)$
Mapping	maps variables into constructions that involve these variables, typically algebraic expressions, for example $f := (x, y, z) \mapsto \sqrt{x^2 + y^2 + z^2}$
Manipulation commands	• To represent function application use (), as in $f(x)$ • To construct expressions, equations and mappings use: =, :=, ->, and unapply to convert an expression into a mapping, • Related to expressions: numer, denom, collect, coeff, degree • Related to equations and inequations: =, <>, <=, >= and to get each side use lhs, rhs, • Basic manipulation of expressions and equations: subs, eval, map, collect, isolate, solve

Table 2: Algebraic expressions, equations and functions

Examples

An algebraic expression

```
> restart;
```

```
> a x^2 + e^x/b
```

$$a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.1)}$$

Note you can think of the labels as names assigned any visible output.

You can also give any name to an expression (assign a name to it) in order to refer to it, and also if the expression is not displayed. You do that by using the assign operator :=

```
> f := (6.1.1)
```

$$f := a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.2)}$$

Now you can refer to the expression **(6.1.1)** using the given name

```
> f
```

$$a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.3)}$$

Different from an expression, an equation always has left and right-hand sides with the `=` operator in between. For example

```
> f = 0
```

(6.1.4)

$$a\,x^2 + \frac{e^x}{b} = 0 \qquad \textbf{(6.1.4)}$$

You get each of the sides using the lhs and rhs commands

> *lhs*(**(6.1.4)**)

$$a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.5)}$$

> *rhs*(**(6.1.4)**)

$$0 \qquad \textbf{(6.1.6)}$$

You can assign names to everything, also to an equation

> $h := f = g$;

$$h := a\,x^2 + \frac{e^x}{b} = g \qquad \textbf{(6.1.7)}$$

What we call "the function of x equal to $a\,x^2 + \frac{e^x}{b}$" is implemented in the computer as a *mapping,* using the arrow operator ->

> $x \rightarrow$ **(6.1.1)**

$$x \mapsto a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.8)}$$

To use a mappings it is also practical to assign a name to it

> $h := x \rightarrow$ **(6.1.1)**

$$h := x \mapsto a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.9)}$$

> $h(x)$

$$a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.10)}$$

> **(6.1.8)**(x)

$$a\,x^2 + \frac{e^x}{b} \qquad \textbf{(6.1.11)}$$

Note however that the *mapping h* not really a function of x, but also of whatever argument you pass to it, as in

> $h(y)$

$$a\,y^2 + \frac{e^y}{b} \qquad \textbf{(6.1.12)}$$

> h(alpha)

$$a\,\alpha^2 + \frac{e^\alpha}{b} \qquad \textbf{(6.1.13)}$$

> **(6.1.8)**(beta)

$$a\,\beta^2 + \frac{e^\beta}{b} \qquad \textbf{(6.1.14)}$$

You can convert an expression or equation into a mapping using unapply

> **(6.1.1)**

$$a\,x^2 + \frac{e^x}{b} \tag{6.1.15}$$

> *unapply*(%, *x*)

$$x \mapsto a\,x^2 + \frac{e^x}{b} \tag{6.1.16}$$

> *unapply*(**(6.1.1)**, *x*, *a*, *b*)

$$(x, a, b) \mapsto a\,x^2 + \frac{e^x}{b} \tag{6.1.17}$$

Returning to the expression *f*

> *f*

$$a\,x^2 + \frac{e^x}{b} \tag{6.1.18}$$

You can get the numerator, denominator or the coefficient of a or of $b^{(-1)}$, or compute the maximum and minimum degrees with respect to any variable

> *numer*(*f*)

$$a\,x^2\,b + e^x \tag{6.1.19}$$

> *denom*(*f*)

$$b \tag{6.1.20}$$

> *coeff*(*f*, *a*)

$$x^2 \tag{6.1.21}$$

> *coeff*(*f*, *b*, −1)

$$e^x \tag{6.1.22}$$

> *degree*(*f*, *b*), *ldegree*(*f*, *b*)

$$0, -1 \tag{6.1.23}$$

> *degree*(*f*, *x*)

$$FAIL \tag{6.1.24}$$

> *degree*$\left(f, e^x\right)$

$$1 \tag{6.1.25}$$

> *frontend*(*degree*, [*f*, *x*])

$$2 \tag{6.1.26}$$

You can substitute into or solve expressions and equations

> *subs*(*x* = 0, *f*)

$$\frac{e^0}{b} \tag{6.1.27}$$

Note the difference with eval: it *evaluates* the function

> *eval*(*f*, *x* = 0)

(6.1.28)

$$\frac{1}{b} \tag{6.1.28}$$

Most functions automatically return a value for their simplest special cases, as e^0. Inert functions are useful to avoid these automatic simplifications, for example:

> $[\%exp(0) = \exp(0), \%sin(0) = \sin(0), \%cos(0) = \cos(0)]$

$$[e^0 = 1, \sin(0) = 0, \cos(0) = 1] \tag{6.1.29}$$

You can activate inert functions using the value command

> $value(\mathbf{(6.1.29)})$

$$[1 = 1, 0 = 0, 1 = 1] \tag{6.1.30}$$

The mathematical properties of the inert functions are understood by the system

> $\%sin(\%cos(z))$

$$\sin(\cos(z)) \tag{6.1.31}$$

> $diff(\mathbf{(6.1.31)}, z)$

$$-\sin(z)\cos(\cos(z)) \tag{6.1.32}$$

You can solve expressions or equations or systems of them. When solving, an expression is considered an equation with right-hand side equal to zero

> f

$$a x^2 + \frac{e^x}{b} \tag{6.1.33}$$

> $isolate(f, b)$

$$b = -\frac{e^x}{a x^2} \tag{6.1.34}$$

> $solve(f, \{b\})$

$$\left\{b = -\frac{e^x}{a x^2}\right\} \tag{6.1.35}$$

isolate however only returns one solution. To get all the solutions use solve

> $isolate(f, x)$

$$x = -2\,W\left(-\frac{\sqrt{-\frac{1}{a\,b}}}{2}\right) \tag{6.1.36}$$

> $solve(f, \{x\})$

$$\left\{x = -2\,W\left(-\frac{\sqrt{-\frac{1}{a\,b}}}{2}\right)\right\}, \left\{x = -2\,W\left(\frac{\sqrt{-\frac{1}{a\,b}}}{2}\right)\right\} \tag{6.1.37}$$

> $lprint(\mathbf{(6.1.37)})$

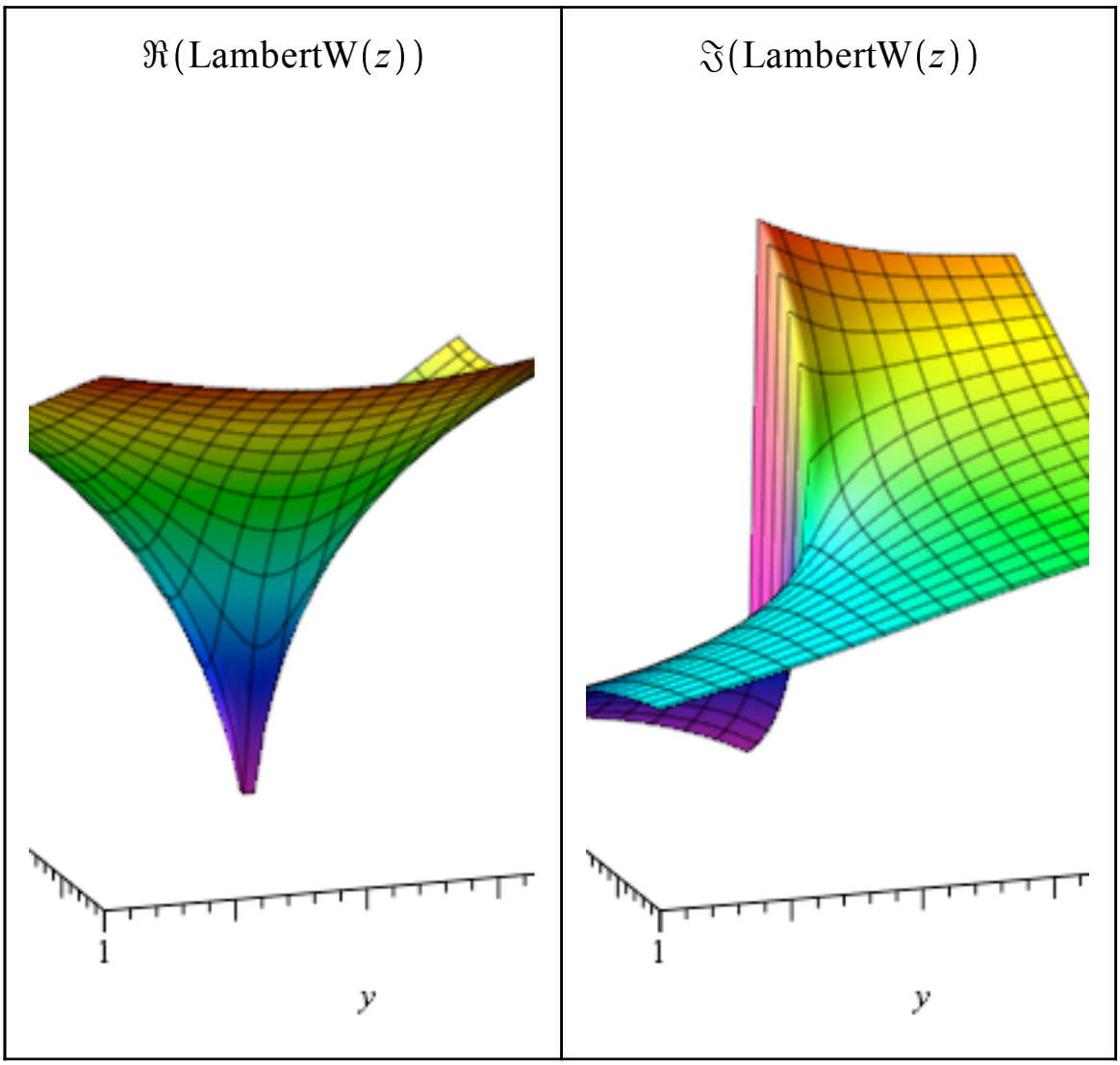
```
{x = -2*LambertW(-(1/2)*(-1/(a*b))^(1/2))}, {x = -2*LambertW(
(1/2)*(-1/(a*b))^(1/2))}
```

> $solve(f, \{x\}, AllSolutions)$

(6.1.38)

$$\left\{x = -2\,W\left(\_Z2\sim, -\frac{\sqrt{-\frac{1}{a\,b}}}{2}\right)\right\}, \left\{x = -2\,W\left(\_Z3\sim, \frac{\sqrt{-\frac{1}{a\,b}}}{2}\right)\right\} \quad \textbf{(6.1.38)}$$

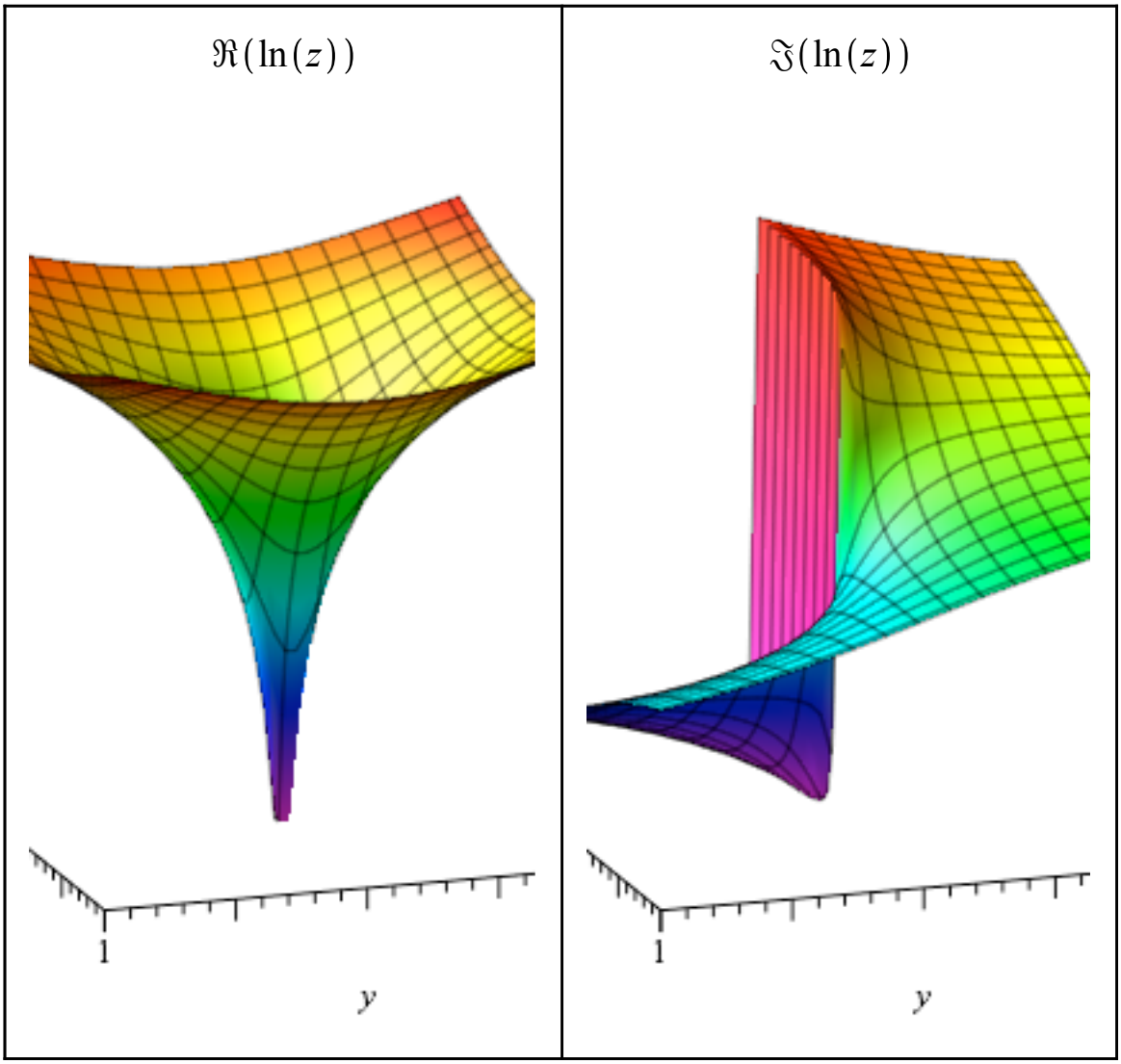
```
> PlotExpression := f → plots:-plotcompare(f, 0, _rest, 'expression_plot', 5) :
> PlotExpression(LambertW(z))
```




```
> FunctionAdvisor(cuts, LambertW)
* Partial match of "cuts" against topic "branch_cuts".
```

$$\left[W(z), z < -\frac{1}{e}\right], \left[W(a, z), a \neq 0 \textbf{ And } z < 0, a = 0 \textbf{ And } z < -\frac{1}{e}\right] \quad \textbf{(6.1.39)}$$

```
> PlotExpression(ln(z))
```



> *FunctionAdvisor*(*def*, LambertW(*z*))
* Partial match of "def" against topic "definition".

$$\left[W(z) = 1 + e^{\frac{\frac{1}{2}\left(\int_0^{\infty} \frac{\ln\left(\frac{\_k1 - I\pi - \ln(\_k1) + \ln(z)}{\_k1 + I\pi - \ln(\_k1) + \ln(z)}\right)}{1 + \_k1}\, d\_k1\right)}{\pi}} (-1 + \ln(z)), \text{And}\left(\textbf{Not } z :: \left[-\frac{1}{e}, 0\right]\right)\right] \quad \textbf{(6.1.40)}$$

>

Exercises

1. Consider what we call $f(x) = \cos(x)^2 + g(x)$

a Enter the expression $\cos(x)^2 + g(x)$

b Use % to refer to this expression and assign the name F to it

c Compute the value of F for: $x = \text{pi}$ and $x = I\,a$

d Transform F into a mapping of x and assign the name G to it

e Use the mapping G to compute the values of item c

Solution

```
> restart
> cos(x)^2 + g(x)
```

$$\cos(x)^2 + g(x) \qquad \textbf{(6.2.1.1)}$$

```
> F := %
```

$$F := \cos(x)^2 + g(x) \qquad \textbf{(6.2.1.2)}$$

```
> eval(F, x = Pi)
```

$$1 + g(\pi) \qquad \textbf{(6.2.1.3)}$$

```
> eval(F, x = I*a)
```

$$\cosh(a)^2 + g(\mathrm{I}\,a) \qquad \textbf{(6.2.1.4)}$$

```
> G := unapply(F, x)
```

$$G := x \mapsto \cos(x)^2 + g(x) \qquad \textbf{(6.2.1.5)}$$

```
> G(Pi)
```

$$1 + g(\pi) \qquad \textbf{(6.2.1.6)}$$

```
> G(I*a)
```

$$\cosh(a)^2 + g(\mathrm{I}\,a) \qquad \textbf{(6.2.1.7)}$$

```
>
```

2. Construct a polynomial of 2nd degree taking the product of monomials of the form $(x - \text{alpha}_j)$ where alpha_j are the roots and compute the maximum and minimum degrees with respect to x, then the coefficients of x to the powers 2,1 and 0, one at a time (coeff) or all at once (coeffs)

Solution

```
> restart
> P := (x - alpha[1]) (x - alpha[2])
```

$$P := (x - \alpha_1)\,(x - \alpha_2) \qquad \textbf{(6.2.2.1)}$$

```
> degree(P, x)
```

$$2 \qquad \textbf{(6.2.2.2)}$$

```
> ldegree(P, x)
```

$$0 \qquad \textbf{(6.2.2.3)}$$

```
> coeff(P, x, 0), coeff(P, x, 1), coeff(P, x, 2)
```

$$\alpha_1\,\alpha_2,\ -\alpha_1 - \alpha_2,\ 1 \qquad \textbf{(6.2.2.4)}$$

```
> seq(coeff(P, x, j), j = [0, 1, 2])
```

$$\alpha_1 \alpha_2,\ -\alpha_1 - \alpha_2,\ 1 \qquad \textbf{(6.2.2.5)}$$

```
> seq(coeff(P, x, j), j = 0..2)
```

$$\alpha_1 \alpha_2,\ -\alpha_1 - \alpha_2,\ 1 \qquad \textbf{(6.2.2.6)}$$

For historical reasons to use 'coeffs' you need to expand the polynomial first - if you don't you hit those non-sense computer idiosyncrasies ...

```
> coeffs(P, x)
Error, invalid arguments to coeffs
```

```
> expand(P)
```

$$x^2 - \alpha_1 x - x \alpha_2 + \alpha_1 \alpha_2 \qquad \textbf{(6.2.2.7)}$$

```
> coeffs(%, x)
```

$$1,\ -\alpha_1 - \alpha_2,\ \alpha_1 \alpha_2 \qquad \textbf{(6.2.2.8)}$$

To avoid these subtleties you can also use the more modern and general Physics:-Coefficients command (also handles anticommutative variables)

```
> Physics:-Coefficients(P, x)
```

$$\alpha_1 \alpha_2,\ -\alpha_1 - \alpha_2,\ 1 \qquad \textbf{(6.2.2.9)}$$

```
>
```

3. Consider the transformation equations between cartesian and spherical coordinates

$$x = r\sin(\text{theta})\cos(\text{phi}),\quad y = r\sin(\text{theta})\sin(\text{phi}),\quad z = r\cos(\text{theta})$$

Use the commands *isolate*, *map* and *subs* - and *assuming* to tell the domain of *r*, theta and phi - in order to invert these equations

Solution

```
> restart
> eq_x := x = r*sin(theta)*cos(phi)
```

$$eq_x := x = r\sin(\theta)\cos(\phi) \qquad \textbf{(6.2.3.1)}$$

```
> eq_y := y = r*sin(theta)*sin(phi)
```

$$eq_y := y = r\sin(\theta)\sin(\phi) \qquad \textbf{(6.2.3.2)}$$

```
> eq_z := z = r*cos(theta)
```

$$eq_z := z = r\cos(\theta) \qquad \textbf{(6.2.3.3)}$$

```
> map(u -> u^2, eq_x)
```

$$x^2 = r^2 \sin(\theta)^2 \cos(\phi)^2 \qquad \textbf{(6.2.3.4)}$$

```
> map(u -> u^2, eq_y)
```

$$y^2 = r^2 \sin(\theta)^2 \sin(\phi)^2 \qquad \textbf{(6.2.3.5)}$$

> $map\left(u \to u^2, eq_z\right)$

$$z^2 = r^2 \cos(\theta)^2 \quad \textbf{(6.2.3.6)}$$

> **(6.2.3.4) + (6.2.3.5) + (6.2.3.6)**

$$x^2 + y^2 + z^2 = r^2 \sin(\theta)^2 \cos(\phi)^2 + r^2 \sin(\theta)^2 \sin(\phi)^2 + r^2 \cos(\theta)^2 \quad \textbf{(6.2.3.7)}$$

> $simplify(\%)$

$$x^2 + y^2 + z^2 = r^2 \quad \textbf{(6.2.3.8)}$$

> $map\left(u \to u^{\frac{1}{2}}, \%\right)$

$$\sqrt{x^2 + y^2 + z^2} = \sqrt{r^2} \quad \textbf{(6.2.3.9)}$$

> $simplify(\%)$ assuming $r > 0$

$$\sqrt{x^2 + y^2 + z^2} = r \quad \textbf{(6.2.3.10)}$$

> $eq_r := (rhs = lhs)(\%)$

$$eq_r := r = \sqrt{x^2 + y^2 + z^2} \quad \textbf{(6.2.3.11)}$$

> $\frac{y}{x} = \frac{rhs\left(eq_y\right)}{rhs\left(eq_x\right)}$

$$\frac{y}{x} = \frac{\sin(\phi)}{\cos(\phi)} \quad \textbf{(6.2.3.12)}$$

> $convert(\%, \tan)$

$$\frac{y}{x} = \tan(\phi) \quad \textbf{(6.2.3.13)}$$

> $eq_{\text{phi}} := isolate(\%, \text{phi})$

$$eq_\phi := \phi = \arctan\left(\frac{y}{x}\right) \quad \textbf{(6.2.3.14)}$$

> $isolate\left(eq_z, \text{theta}\right)$

$$\theta = \arccos\left(\frac{z}{r}\right) \quad \textbf{(6.2.3.15)}$$

> $eq_{\text{theta}} := subs\left(eq_r, \%\right)$

$$eq_\theta := \theta = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \quad \textbf{(6.2.3.16)}$$

>

3. Limits, Derivatives, Sums, Products, Integrals, Differential Equations

Commands	limit, diff and D, sum, product, int, dsolve, pdsolve
Manipulation commands	PDEtools:-dchange, PDEtools:-casesplit, the inert forms %limit, %int, etc. and the related value command

Table 3: Calculus

Examples

The commands to compute limits, derivatives, sums and products are limit, diff, sum, product. The D command also represents derivatives - more about this afterwards.

```
> restart;
```

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

$$1 \qquad \textbf{(7.1.1)}$$

All Maple commands have an inert version of them, that represent the mathematical object but does not perform the computation until you require it using the value command. Inert subexpressions always have some part displayed in grey:

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

$$\lim_{x \to 0} \frac{\sin(x)}{x} \qquad \textbf{(7.1.2)}$$

```
> value((7.1.2))
```

$$1 \qquad \textbf{(7.1.3)}$$

```
> (d/dx) g(x) + exp(x^2)
```

$$\frac{d}{dx}\left(g(x) + e^{x^2}\right) \qquad \textbf{(7.1.4)}$$

```
> value((7.1.4))
```

$$g'(x) + 2\,x\,e^{x^2} \qquad \textbf{(7.1.5)}$$

Handy: functionality is distributed over the sides of equations, so you can write this:

```
%sum(x^n/n!, n = 0 ..infinity) = sum(x^n/n!, n = 0 ..infinity)
```

directly as

```
> (%sum = sum)(x^n/n!, n = 0 ..infinity)
```

$$\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x \qquad \textbf{(7.1.6)}$$

All the family of *sum*, *int*, *solve*, *dsolve* and *pdsolve* are rather powerful commands nowadays. In

the case of summation, note that it can also be performed in the indefinite case with the meaning:

$$\sum_k f(k) = g(k) \quad \text{where} \quad g(k+1) - g(k) = f(k)$$

When entering the following command, you will be asked whether it represents a function definition or a remember table assignment, choose remember table assignment (to perform these assignments with a function on the left-hand side without being asked questions enter first you can also enter *Typesetting*:-*Settings*('*functionassign* = *false*'))

> $f(k) := \left(\frac{k}{2}\right)! \, k$

$$f(k) := \left(\frac{k}{2}\right)! \, k \tag{7.1.7}$$

> *%sum*$(f(k), k)$;

$$\sum_k \left(\frac{k}{2}\right)! \, k \tag{7.1.8}$$

> *value*(**(7.1.8)**)

$$2\left(\frac{k}{2}\right)! + 2\left(\frac{k}{2} + \frac{1}{2}\right)! \tag{7.1.9}$$

> *eval*(**(7.1.9)**, $k = \mathrm{k} + 1$) - **(7.1.9)**

$$2\left(\frac{k}{2} + 1\right)! - 2\left(\frac{k}{2}\right)! \tag{7.1.10}$$

> *simplify*(*%*)

$$\left(\frac{k}{2}\right)! \, k \tag{7.1.11}$$

Results are frequently expressed in terms of not-so-familiar special functions

> *%int*$\left(\mathrm{e}^{-x^2}, x\right)$;

$$\int \mathrm{e}^{-x^2} \, \mathrm{d}x \tag{7.1.12}$$

> *value*(**(7.1.12)**)

$$\frac{\sqrt{\pi}\ \mathrm{erf}(x)}{2} \tag{7.1.13}$$

> *%int*$\left(\frac{1}{\sqrt{2t^4 - 3t^2 - 2}}, t = 2..3\right)$

$$\int_2^3 \frac{1}{\sqrt{2t^4 - 3t^2 - 2}} \, \mathrm{d}t \tag{7.1.14}$$

> *value*(**(7.1.14)**)

$$\frac{\sqrt{5}\, F\left(\frac{\sqrt{7}}{3}, \frac{\sqrt{5}}{5}\right)}{5} - \frac{\sqrt{5}\, F\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{5}}{5}\right)}{5} \tag{7.1.15}$$

> *lprint*(%)

```
(1/5)*5^(1/2)*EllipticF((1/3)*7^(1/2), (1/5)*5^(1/2))-(1/5)*
5^(1/2)*EllipticF((1/2)*2^(1/2), (1/5)*5^(1/2))
```

Most of these commands have options to workaround special cases

> $int\left(\frac{1}{x}, x = a..2\right)$

Warning, unable to determine if 0 is between a and 2; try to use assumptions or use the AllSolutions option

$$\int_a^2 \frac{1}{x}\,dx \tag{7.1.16}$$

> $int\left(\frac{1}{x}, x = a..2, 'AllSolutions'\right)$

$$\begin{cases} undefined & a < 0 \\ \infty & a = 0 \\ -\ln(a) + \ln(2) & 0 < a \end{cases} \tag{7.1.17}$$

The *assuming* command is also handy in these cases

> **(7.1.16)** assuming $a > 0$;

$$-\ln(a) + \ln(2) \tag{7.1.18}$$

> **(7.1.16)** assuming $a < 0$;

$$undefined \tag{7.1.19}$$

The ordinary and partial differential equation commands have by now concentrated so much solving power that themselves are used to develop new solving algorithms

> *PDEtools*:-*declare*$(y(x), prime = x)$

$$y(x) \text{ will now be displayed as } y$$

$$\text{derivatives with respect to x of functions of one variable will now be displayed with '} \tag{7.1.20}$$

> $ode_2 := y'(x) - y(x)^2 + (y(x)\sin(x)) - \cos(x) = 0$

$$ode_2 := y' - y^2 + y\sin(x) - \cos(x) = 0 \tag{7.1.21}$$

> *dsolve*$(ode[2])$

$$y = -\frac{e^{-\cos(x)}}{\_C1 + \int e^{-\cos(x)}\,dx} + \sin(x) \tag{7.1.22}$$

> $ode_3 := y'(x) = \frac{x\left(-x - 1 + x^2 - 2x^2 y(x) + \left(2x^4\right)\right)}{\left(\left(x^2 - y(x)\right)(x+1)\right)}$

$$ode_3 := y' = \frac{x\left(x^2 - x - 1 - 2x^2 y + 2x^4\right)}{\left(x^2 - y\right)(x+1)} \tag{7.1.23}$$

Computing an integrating factor

> *DEtools*[*intfactor*]$(ode[3])$

$$\frac{-x^2 + y}{-2x^2 + 2y - 1} \tag{7.1.24}$$

The product of an ode and its integrating factor results in an total derivative

> **(7.1.24) (7.1.23)**

$$\frac{\left(-x^2+y\right)y'}{-2x^2+2y-1}=\frac{\left(-x^2+y\right)x\left(x^2-x-1-2x^2y+2x^4\right)}{\left(-2x^2+2y-1\right)\left(x^2-y\right)(x+1)} \tag{7.1.25}$$

From where

> *dsolve*(**(7.1.25)**)

$$y=-\frac{1}{2(x+1)^4}\left(-2x^6-8x^5-13x^4-12x^3-8x^2+e^{-W\left(-\frac{e^{\frac{4x^3}{3}}(e^x)^4e^{-1}}{(e^{x^2})^4(e^{\_C1})^4(x+1)^4}\right)+\frac{4x^3}{3}-4x^2-4\_C1+4x-1}-4x-1\right) \tag{7.1.26}$$

> *simplify*(**(7.1.26)**)

$$y=x^2+\frac{W\left(-\frac{e^{\frac{4}{3}x^3-4x^2-4\_C1+4x-1}}{(x+1)^4}\right)}{2}+\frac{1}{2} \tag{7.1.27}$$

The notation for special functions is frequently unfamiliar - use lprint

> *lprint*(**(7.1.27)**)

```
y(x) = x^2+(1/2)*LambertW(-exp((4/3)*x^3-4*x^2-4*_C1+4*x-1)/
(x+1)^4)+1/2
```

Laplace equation in spherical coordinates:

> *PDEtools*:-*declare*(*F*(*r*, theta, phi))

$$F(r,\theta,\phi)\ \textit{will now be displayed as}\ F \tag{7.1.28}$$

> $$PDE:=\frac{d}{dr}r^2\left(\frac{\partial}{\partial r}F(r,\text{theta},\text{phi})\right)+\frac{\frac{d}{d\text{theta}}\sin(\text{theta})\left(\frac{\partial}{\partial\text{theta}}F(r,\text{theta},\text{phi})\right)}{\sin(\text{theta})}+\frac{\left(\frac{\partial^2}{\partial\text{phi}^2}F(r,\text{theta},\text{phi})\right)}{\sin(\text{theta})^2}=0$$

$$PDE:=\frac{\partial}{\partial r}\left(r^2F_r\right)+\frac{\frac{\partial}{\partial\theta}\left(\sin(\theta)F_\theta\right)}{\sin(\theta)}+\frac{F_{\phi,\phi}}{\sin(\theta)^2}=0 \tag{7.1.29}$$

> *value*(*PDE*)

$$2rF_r+r^2F_{r,r}+\frac{\cos(\theta)F_\theta+\sin(\theta)F_{\theta,\theta}}{\sin(\theta)}+\frac{F_{\phi,\phi}}{\sin(\theta)^2}=0 \tag{7.1.30}$$

The standard solution separating variables by product

> *pdsolve*(*PDE*)

$$\left(F = \_F1(r)\,\_F2(\theta)\,\_F3(\phi)\right) \text{ \&where } \left[\left\{\_F1_{r,r} = \frac{\_F1(r)\,\_c_1}{r^2} - \frac{2\,\_F1_r}{r}, \_F2_{\theta,\theta} = -\_F2(\theta)\,\_c_1 + \frac{\_F2(\theta)\,\_c_2}{\sin(\theta)^2} - \frac{\cos(\theta)\,\_F2_\theta}{\sin(\theta)}, \_F3_{\phi,\phi} = -\_F3(\phi)\,\_c_2\right\}\right] \quad \textbf{(7.1.31)}$$

Laplace equation also admits a solution separable by sum

> *pdsolve*(*PDE*, *HINT* = `+`)

$$\left(F = \_F1(r) + \_F2(\theta) + \_F3(\phi)\right) \text{ \&where } \left[\left\{\_F1_{r,r} = \frac{\_c_1}{r^2} - \frac{2\,\_F1_r}{r}, \_F2_{\theta,\theta} = -\_c_1 - \frac{\cos(\theta)\,\_F2_\theta}{\sin(\theta)} - \frac{\_c_3}{\sin(\theta)^2}, \_F3_{\phi,\phi} = \_c_3\right\}\right] \quad \textbf{(7.1.32)}$$

You transform these structures into a concrete solution using the build command, or using the build option as in pdsolve(PDE, build). For example

> *PDEtools*:-*build*(%)

$$\begin{aligned} F = {} & -\frac{\_c_3 \ln\left(\frac{I\left(-e^{I\theta}+1\right)}{e^{I\theta}+1}\right)^2}{2} + \_c_1 \ln(r) + \_C3 \ln\left(\frac{I\left(-e^{I\theta}+1\right)}{e^{I\theta}+1}\right) \\ & + \_c_1 \ln\left(\frac{I\left(-e^{I\theta}+1\right)}{e^{I\theta}+1}\right) - 2\,\_c_1 \ln(2) - \frac{\_C3 \ln\left(\frac{e^{I\theta}}{\left(e^{I\theta}+1\right)^2}\right)}{2} \\ & - \frac{\_c_1 \ln\left(\frac{e^{I\theta}}{\left(e^{I\theta}+1\right)^2}\right)}{2} + \frac{\_C3 \ln\left(-\frac{e^{I\theta}}{\left(e^{I\theta}+1\right)^2}\right)}{2} \\ & - \frac{\_c_1 \ln\left(-\frac{e^{I\theta}}{\left(e^{I\theta}+1\right)^2}\right)}{2} + \frac{\phi^2\,\_c_3}{2} + \_C5\,\phi + \_C6 + \_C2 + \_C4 - \frac{\_C1}{r} \end{aligned} \quad \textbf{(7.1.33)}$$

Symmetry methods and systems of partial differential equations, linear and nonlinear, can be solved in many cases.

A nonlinear ODE and the linear PDE system for its symmetries

> $ode_{11} := y''(x) + \left(a\,x^r\,y(x)^n\right) = 0$

$$ode_{11} := y'' + a\,x^r\,y^n = 0 \quad \textbf{(7.1.34)}$$

> *sys* := [*DEtools*[*gensys*](*ode*[11], [xi, eta](*x*, *y*))];

(7.1.35)

$$sys := \left[\xi_{y,y},\ -2\,\xi_{x,y} + \eta_{y,y},\ 3\,\xi_y\,x^r\,y^n\,a + 2\,\eta_{x,y} - \xi_{x,x},\ 2\,\xi_x\,x^r\,y^n\,a - \eta_y\,x^r\,y^n\,a + \frac{\eta(x,y)\,a\,x^r\,y^n\,n}{y} + \frac{\xi(x,y)\,a\,x^r\,r\,y^n}{x} + \eta_{x,x}\right] \quad (7.1.35)$$

> $pdsolve(sys)$

$$\left\{\eta(x,y) = -\frac{\_C1\,y\,(r+2)}{n-1},\ \xi(x,y) = \_C1\,x\right\} \quad (7.1.36)$$

A solution for this system such that r is a parameter (so, also a solving variable) and n is different from 1

> $sys_2 := [DEtools[gensys](ode[11], [\mathrm{xi}, \mathrm{eta}](x,y)),\ n \neq 1];$

$$sys_2 := \left[\xi_{y,y},\ -2\,\xi_{x,y} + \eta_{y,y},\ 3\,\xi_y\,x^r\,y^n\,a + 2\,\eta_{x,y} - \xi_{x,x},\ 2\,\xi_x\,x^r\,y^n\,a - \eta_y\,x^r\,y^n\,a + \frac{\eta(x,y)\,a\,x^r\,y^n\,n}{y} + \frac{\xi(x,y)\,a\,x^r\,r\,y^n}{x} + \eta_{x,x},\ n \neq 1\right] \quad (7.1.37)$$

> $pdsolve(sys_2, [\mathrm{xi}, \mathrm{eta}, r])$

$$\{r = -2,\ \eta(x,y) = 0,\ \xi(x,y) = \_C1\,x\},\ \{r = -n-3,\ \eta(x,y) = 0,\ \xi(x,y) = 0\},\ \left\{r = -n-3,\ \eta(x,y) = (\_C1\,x + \_C2)\,y,\ \xi(x,y) = \frac{(\_C1\,x + (\_C1\,x + \_C2)\,n - \_C2)\,x}{n+1}\right\},\ \left\{r = r,\ \eta(x,y) = \_C1\,y,\ \xi(x,y) = -\frac{\_C1\,x\,(n-1)}{r+2}\right\} \quad (7.1.38)$$

Indeed if you take r as a solving variable, using differential algebra techniques the problem splits into three different problems (so called: the general and the singular cases:

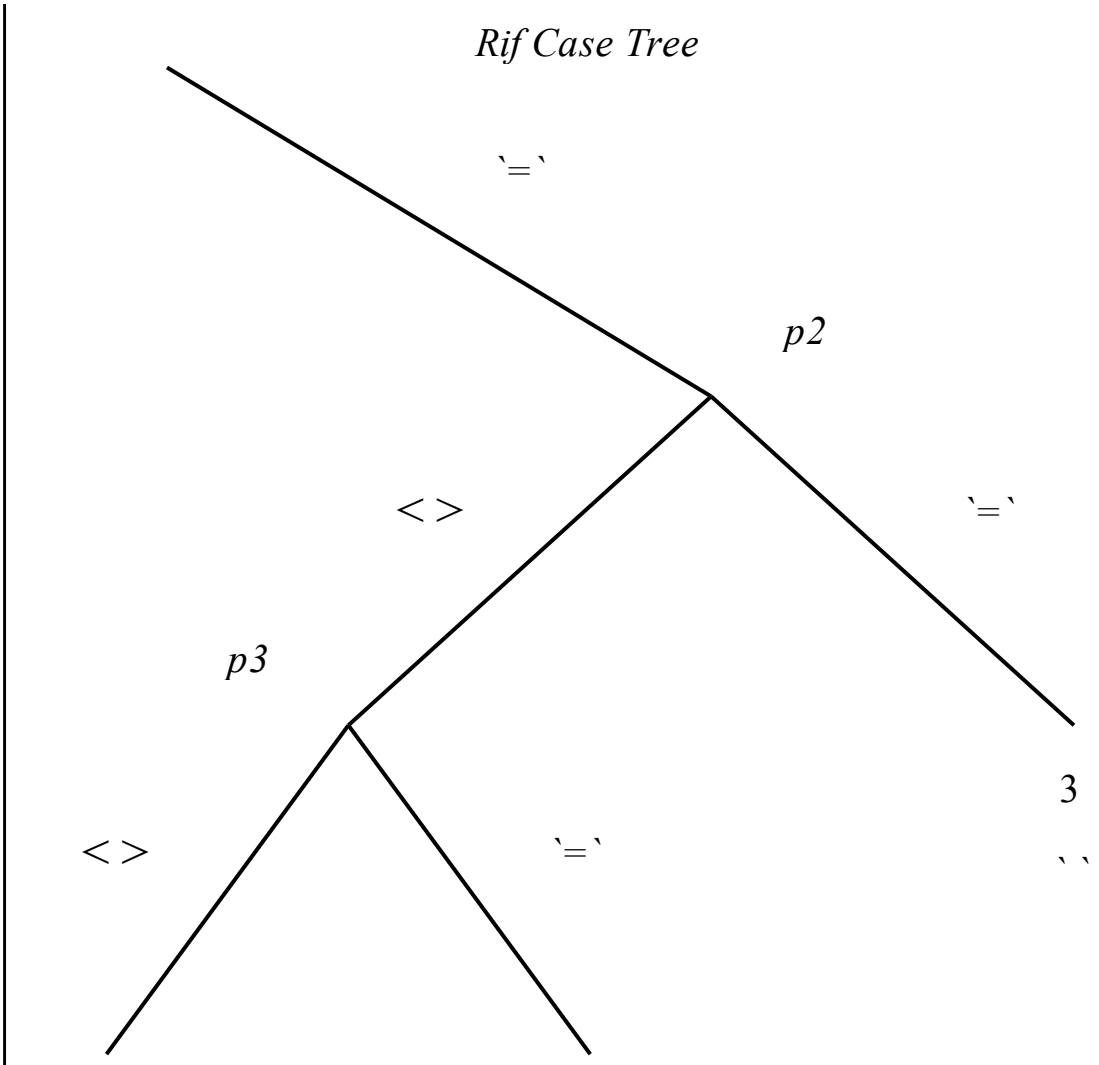
> $PDEtools:\text{-}casesplit(sys_2, [\mathrm{xi}, \mathrm{eta}, r], caseplot)$

$$========= \textit{Pivots Legend} =========$$

$$p1 = \eta_{y,x,y}$$

$$p2 = 2\,x^r + (x^r)_x\,x$$

$$p3 = -(x^r)_x\,x - x^r\,(n+3)$$



$$\left[\xi(x,y)=\frac{-\eta(x,y)\,n\,x+\eta(x,y)\,x}{r\,y+2\,y},\ \eta_x=0,\ \eta_y=\frac{\eta(x,y)}{y}\right]\ \&\text{where}\ \left[(r+2)\,x^r\neq 0,\ -(r+n+3)\,x^r\neq 0\right],\left[\xi(x,y)=\frac{2\,\eta_x\,x^2+\eta(x,y)\,n\,x-\eta(x,y)\,x}{n\,y+y},\ \eta_{x,x}=0,\ \eta_y=\frac{\eta(x,y)}{y},\ r=-n-3\right]\ \&\text{where}\ [\ ],\left[\xi_x=\frac{\xi(x,y)}{x},\ \xi_y=0,\ \eta(x,y)=0,\ r=-2\right]\ \&\text{where}\ [\ ] \quad \textbf{(7.1.39)}$$

>

Exercises

There is no much to say about all these commands but for dsolve and pdsolve, the solvers for ordinary and partial differential equations, as well as the PDEtools:-casesplit command for triangularizing systems of equations. So the exercises for this section are about exploration.

1. Open the help page for dsolve/education ,
   a. Transform the page into a worksheet (one of the icons on the toolbar);

b. Go to the menu View -> Collapse All Sections, and choose a section you want to explore. My recommendation according to how useful it could be in physics computations:

* If you are not familiar with symmetry methods, the corresponding section can give you a rapid glimpse on how to tackle ODEs by discovering their symmetries and using them to construct solutions

* The section on singular solutions may open your eyes to something you are probably not aware of regarding differential equations. Singular solutions are frequently the ones that are relevant in physics models. The relevant command here is PDEtools:-casesplit

* The section on using "this or that method" has no mathematical insights but is useful information regarding flexibility for computing different forms of general ODE solutions

---

2. The same with the help page for PDEtools:-InvariantSolutions

---

3. The product of an integrating factor an a differential equation is a *Total Derivative*. Use DEtools[redode] to construct a second order ODE family having an integrating factor mu = $F(x)$ -- an arbitrary function -- such that the reduced ODE has the same integrating factor.

Solution

This problem is solved as the first part of problem 1. in the help page for DEtools[redode]

4. An ODE of order n^{th} admits n integrating factors. Use DEtools[redode] to construct the most general third order ODE admitting the
following three integrating factors:

$$\left[y(x)\ x, x^{-\frac{1}{4}+\frac{1}{4}\sqrt{29}}\ y(x), x^{-\frac{1}{4}-\frac{1}{4}\sqrt{29}}\ y(x)\right]$$

then use DEtools[mutest] to verify that the obtained ODE admits these 3 expressions as integrating factors

Solution

This problem is solved as the first part of problem 1. in the help page for DEtools[redode]

```
> mu := [y(x)*x, x^(-1/4+(1/4)*29^(1/2))*y(x), x^(-1/4-(1/4)*29^(1/2))*y(x)]
```

$$\mu := \left[y\,x, x^{-\frac{1}{4}+\frac{\sqrt{29}}{4}}\ y, x^{-\frac{1}{4}-\frac{\sqrt{29}}{4}}\ y\right] \qquad \textbf{(7.2.2.1)}$$

```
> DEtools[redode](mu, 3, y(x))
```

$$y''' = \frac{13\,y}{8\,x^3} + \frac{5\,y''}{2\,x} - \frac{13\,y'}{4\,x^2} + \frac{-3\,y'\,y'' + \frac{5\,y^2}{2\,x} + \_F1(x)}{y} \qquad \textbf{(7.2.2.2)}$$

```
> map(DEtools[mutest], mu, (7.2.2.2))
```

$$[0, 0, 0] \qquad \textbf{(7.2.2.3)}$$

>

4. Algebraic manipulation: simplify, factor, expand, combine, collect and convert

Commands	simplify, factor, expand, combine, collect and convert

Table 4: Algebraic manipulation

Examples

Simplification is not really a well defined operation, but one based on common sense, and the desired result sometimes depends on particularities of the problem.

Among the most typical simplifications there is the one that *makes use of functions identities*

> $\sin(x)^2 + \cos(x)^2$

$$\sin(x)^2 + \cos(x)^2 \quad \textbf{(8.1.1)}$$

> *simplify*(**(8.1.1)**)

$$1 \quad \textbf{(8.1.2)}$$

Another typical simplification is the *simplification in size*

> $\frac{e^{-\frac{1x^2}{4}}\, 2^{\frac{1}{4}}\, x^{\frac{3}{2}}}{4} + \frac{1\, e^{\frac{1x^2}{4}}\, 2^{\frac{3}{4}}\, \sqrt{x}\, \sqrt{\mathrm{pi}}\, \mathrm{F}(x)}{8} + \frac{1\, e^{\frac{1x^2}{4}}\, 2^{\frac{3}{4}}\, x^{\frac{5}{2}}\, \sqrt{\mathrm{pi}}\, \mathrm{F}(x)}{8}$

$$\frac{e^{-\frac{x^2}{4}}\, 2^{1/4}\, x^{3/2}}{4} + \frac{e^{\frac{x^2}{4}}\, 2^{3/4}\, \sqrt{x}\, \sqrt{\pi}\, F(x)}{8} + \frac{e^{\frac{x^2}{4}}\, 2^{3/4}\, x^{5/2}\, \sqrt{\pi}\, F(x)}{8} \quad \textbf{(8.1.3)}$$

> *simplify*(*%*, *size*)

$$\frac{\sqrt{x}\left(\sqrt{2}\, \sqrt{\pi}\, (x^2+1)\, F(x)\, e^{\frac{x^2}{4}} + 2\, e^{-\frac{x^2}{4}}\, x\right) 2^{1/4}}{8} \quad \textbf{(8.1.4)}$$

In other cases, it all depends on what is preferred

> $6\,(x+4)\,(x-1)$

$$6\,(x+4)\,(x-1) \quad \textbf{(8.1.5)}$$

> $6x^2 + 18x - 24$

$$6x^2 + 18x - 24 \quad \textbf{(8.1.6)}$$

These two expressions are equal, and both are 'simplified', so simplify does nothing

> *simplify*(**(8.1.5)**)

$$6\,(x+4)\,(x-1) \quad \textbf{(8.1.7)}$$

> *simplify*(**(8.1.6)**)

$$6\,x^2 + 18\,x - 24 \qquad \textbf{(8.1.8)}$$

To rewrite one as the other one, the operations to be performed are: to *factor* or to *expand*

> *factor*(**(8.1.5)**)

$$6\,(x+4)\,(x-1) \qquad \textbf{(8.1.9)}$$

> *expand*(**(8.1.9)**)

$$6\,x^2 + 18\,x - 24 \qquad \textbf{(8.1.10)}$$

In this case the expanded form is also a form where powers of x are collected, as in

> *collect*(**(8.1.9)**, x)

$$6\,x^2 + 18\,x - 24 \qquad \textbf{(8.1.11)}$$

One of the most powerful simplifications is to simplify with respect to given equations, for example: "simplify $6\,x^2 + 18\,x - 24$ taking x+4=alpha and x-1=beta

> *simplify*(**(8.1.11)**, $\{x + 4 = \text{alpha}\}$)

$$6\,\alpha^2 - 30\,\alpha \qquad \textbf{(8.1.12)}$$

Both expand and combine take into account the properties of mathematical functions, with the combine command rewriting powers of trigonometric functions as expressions linear in other trigonometric functions

> $\sin(x)^2 - \cos(x)^2$

$$\sin(x)^2 - \cos(x)^2 \qquad \textbf{(8.1.13)}$$

> *combine*(**(8.1.13)**)

$$-\cos(2\,x) \qquad \textbf{(8.1.14)}$$

> $\%sum\left(a[j]\cdot\cos(j\cdot x)^j + b[j]\,\sin(j\,x)^{2\,j}, j = 1\,..2\right)$

$$\sum_{j=1}^{2}\left(a_j\,\cos(j\,x)^j + b_j\,\sin(j\,x)^{2\,j}\right) \qquad \textbf{(8.1.15)}$$

> *value*(**(8.1.15)**)

$$a_1\,\cos(x) + b_1\,\sin(x)^2 + a_2\,\cos(2\,x)^2 + b_2\,\sin(2\,x)^4 \qquad \textbf{(8.1.16)}$$

> *combine*(**(8.1.16)**)

$$a_1\,\cos(x) - \frac{b_1\,\cos(2\,x)}{2} + \frac{b_1}{2} + \frac{a_2\,\cos(4\,x)}{2} + \frac{a_2}{2} - \frac{b_2\,\cos(4\,x)}{2} + \frac{b_2\,\cos(8\,x)}{8} + \frac{3\,b_2}{8} \qquad \textbf{(8.1.17)}$$

We almost always want to 'simplify in size':

> *simplify*(**(8.1.17)**, *size*)

$$\frac{(4\,a_2 - 4\,b_2)\,\cos(4\,x)}{8} + a_1\,\cos(x) - \frac{b_1\,\cos(2\,x)}{2} + \frac{b_2\,\cos(8\,x)}{8} + \frac{a_2}{2} + \frac{b_1}{2} + \frac{3\,b_2}{8} \qquad \textbf{(8.1.18)}$$

In this following example the simplification in size is more convenient that a direct simplification

> $e2 := -\left(3\sin(x)^{\frac{1}{2}}\cos(x)^2\sin(x)^m\right) + \left(3\sin(x)^{\frac{1}{2}}\cos(x)^2\cos(x)^n\right) + \left(4\sin(x)^{\frac{1}{2}}\cos(x)^4\sin(x)^m\right) - 4\sin(x)^{\frac{1}{2}}\cos(x)^4\cos(x)^n$

$$e2 := -3\sqrt{\sin(x)}\cos(x)^2\sin(x)^m + 3\sqrt{\sin(x)}\cos(x)^2\cos(x)^n + 4\sqrt{\sin(x)}\cos(x)^4\sin(x)^m - 4\sqrt{\sin(x)}\cos(x)^4\cos(x)^n \quad \textbf{(8.1.19)}$$

> $simplify(e2, size)$

$$-4\left(\cos(x)^2 - \frac{3}{4}\right)\sqrt{\sin(x)}\left(\cos(x)^n - \sin(x)^m\right)\cos(x)^2 \quad \textbf{(8.1.20)}$$

> $simplify(e2)$

$$\cos(x)^2\sqrt{\sin(x)}\left(4\cos(x)^2\sin(x)^m - 4\cos(x)^{2+n} - 3\sin(x)^m + 3\cos(x)^n\right) \quad \textbf{(8.1.21)}$$

Other times what we really want is not a *simplification* but to have an expression rewritten with powers of same variables factored out (we say 'collected'), for example

> $p := b\,y + 6\,g\,x + x\,y$

$$p := b\,y + 6\,g\,x + y\,x \quad \textbf{(8.1.22)}$$

> $collect(p, [x, y])$

$$(6\,g + y)\,x + b\,y \quad \textbf{(8.1.23)}$$

> $collect(p, [y, x])$

$$(b + x)\,y + 6\,g\,x \quad \textbf{(8.1.24)}$$

Sometimes all what we want is to cancel factors that appear in the numerator and denominator of an expressions, for example:

> $num := expand((x - a)\,(x - b))$

$$num := a\,b - a\,x - x\,b + x^2 \quad \textbf{(8.1.25)}$$

> $den := expand((x - a)\,(x - c))$

$$den := a\,c - a\,x - x\,c + x^2 \quad \textbf{(8.1.26)}$$

> $\frac{num}{den}$

$$\frac{a\,b - a\,x - x\,b + x^2}{a\,c - a\,x - x\,c + x^2} \quad \textbf{(8.1.27)}$$

To cancel common factors in the numerator and denominator we use *normal*

> $normal(\textbf{(8.1.27)})$

$$\frac{b - x}{c - x} \quad \textbf{(8.1.28)}$$

Finally what we sometimes want is just to rewrite an expression in terms of different functions, for example

> $\frac{1}{4}\left(e^x\right)^2 - \frac{\frac{1}{4}}{\left(e^x\right)^2}$

(8.1.29)

$$\frac{\left(e^{x}\right)^{2}}{4}-\frac{1}{4\left(e^{x}\right)^{2}} \tag{8.1.29}$$

> *simplify*(**(8.1.29)**)

$$\frac{e^{2x}}{4}-\frac{e^{-2x}}{4} \tag{8.1.30}$$

> *convert*(**(8.1.29)**, *trig*)

$$\frac{(\cosh(x)+\sinh(x))^{2}}{4}-\frac{1}{4(\cosh(x)+\sinh(x))^{2}} \tag{8.1.31}$$

> *simplify*(**(8.1.31)**)

$$\cosh(x)\sinh(x) \tag{8.1.32}$$

You can try converting any function into any other one, and the conversion will (almost always) proceed when the conversion is possible

>

Exercises

1. Show, algebraically, using *simplify* and *assuming*, that $\sqrt{z^2}=z$ when z is real and positive and discover the most general domain for z such that the identity holds

Solution

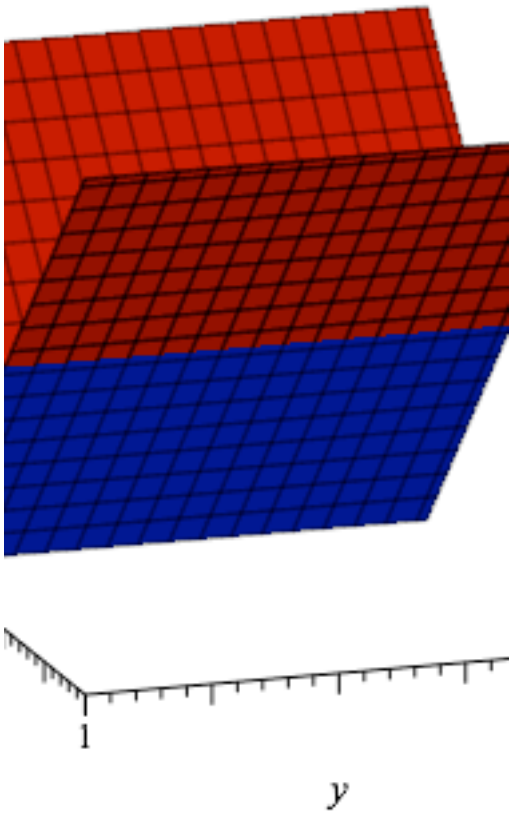
> sqrt$\left(z^2\right)=z$

$$\sqrt{z^2}=z \tag{8.2.1.1}$$

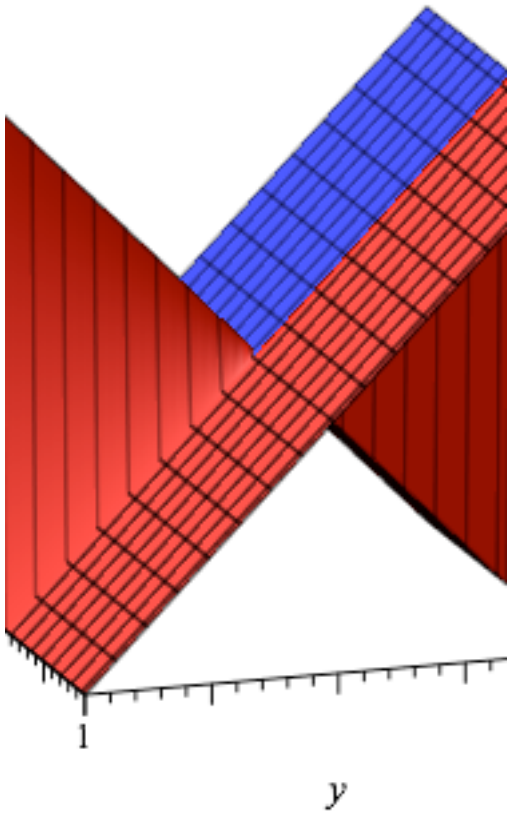
You can see this expression is not equal to z by comparing both expressions using plots[plotcompare]

> *plots*[*plotcompare*](**(8.2.1.1)**, *same_box*)

$\Re\left(\sqrt{(x+\mathrm{I}\,y)^2}\right)$ **and** $\Re(x+\mathrm{I}\,y)$



$\Im\left(\sqrt{(x+\mathrm{I}\,y)^2}\right)$ **and** $\Im(x+\mathrm{I}\,y)$



Rotating the plots you see that these two functions are equal when $0 < z$. To simplify algebraically assuming that z > 0 , use

> *simplify*(**(8.2.1.1)**) assuming $z > 0$;

$$z = z \qquad \textbf{(8.2.1.2)}$$

To discover the most general domain for z such that the identity holds, by trial an error the first thing one could do is to try simplifying the expression as given:

> *simplify*(**(8.2.1.1)**)

$$\mathrm{csgn}(z)\,z = z \qquad \textbf{(8.2.1.3)}$$

and, from its help page, csgn(z) is equal to 1 only when
$0 < \Re(z)$ **or** $(\Re(z) = 0$ **and** $0 < \Im(z))$.

So the following also simplifies to an identity

> *simplify*(**(8.2.1.1)**) assuming $\Re(z) = 0$ **and** $0 < \Im(z)$

$$z = z \qquad \textbf{(8.2.1.4)}$$

>

2. Use a *simplification taking into account that* $\sin^2 + \cos^2 = 1$ (see simplify,siderels) to show that

$$8\sin(x)^4\cos(x) + 15\sin(x)^2\cos(x)^3 - 15\sin(x)^2\cos(x) + 7\cos(x)^5 - 14\cos(x)^3 + 7\cos(x)$$

is equal to 0.

Solution

```
> f := 8*sin(x)^4*cos(x) + 15*sin(x)^2*cos(x)^3 - 15*sin(x)^2*cos(x) + 7*cos(x)^5
       - 14*cos(x)^3 + 7*cos(x)
```

$$f := 8\sin(x)^4\cos(x) + 15\sin(x)^2\cos(x)^3 - 15\sin(x)^2\cos(x) + 7\cos(x)^5 - 14\cos(x)^3 + 7\cos(x) \quad \textbf{(8.2.2.1)}$$

```
> eq := {sin(x)^2 + cos(x)^2 = 1}
```

$$eq := \{\sin(x)^2 + \cos(x)^2 = 1\} \quad \textbf{(8.2.2.2)}$$

```
> simplify(f, eq)
```

$$0 \quad \textbf{(8.2.2.3)}$$

```
>
```

5. Matrices (Linear Algebra)

Commands	Matrix, Vector is the same as Vector[colum], Vector[row], or matrix and vector. Use + and . for operations
Manipulation commands	LinearAlgebra package: conjugate, Transpose, HermitianTranspose, Determinant, Trace, Eigenvalues, Eigenvectors, MatrixExponential, LinearSolve linalg package: conjugate, transpose, htranspose, det, trace, eigenvalues, eigenvectors, exponential, linsolve

Table 5: Linear Algebra

Examples

There is a whole LinearAlgebra package with 130 commands to manipulate Matrices and solve linear algebra problems.
There is also the older linalg package with 114 matrix algebra commands.

Here we restrict to a small subset of matrix commands that are used more frequently, and for the rest: just consult help pages when necessary.

For historical and other reasons, there are two kind of matrices in Maple.

- The old ones, represented by the lowercase word matrix have the advantage that you can compute with them without displaying their contents.
- The new ones, represented by the word Matrix have the advantage of performing component computations faster

First *matrix*

```
> A := matrix(2, 2, [a, b, c, d])
```

$$A := \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \textbf{(9.1.1)}$$

Invoking the matrix does not show its components

```
> A
```

$$A \quad \textbf{(9.1.2)}$$

You can refer to an unspecified component (this is useful when setting brackets rules in Quantum Mechanics), as in

```
> A[i,j]
```

$$A_{i,j} \quad \textbf{(9.1.3)}$$

You any specified component by attributing values to the indices

```
> eval((9.1.3), [i = 1, j = 2])
```

$$b \quad \textbf{(9.1.4)}$$

The same with *Matrix*

```
> B := Matrix(2, 2, [a, b, c, d])
```

$$B := \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \textbf{(9.1.5)}$$

Invoking it shows its components

```
> B
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \textbf{(9.1.6)}$$

You cannot refer to an unspecified component

```
> B[i,j]
Error, bad index into Matrix
```

The *LinearAlgebra* package is all about *Matrix*, while there also exists the old *linalg* package about *matrix*. So you can do operations with both packages according to whether you need more symbolic capabilities (linalg) or faster computations (LinearAlgebra).

There are routines to convert a *matrix* into a *Matrix* and the other way around

```
> C := convert(B, matrix)
```

$$C := \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \textbf{(9.1.7)}$$

```
> C
```

$$C \tag{9.1.8}$$

> $C[i,j]$

$$C_{i,j} \tag{9.1.9}$$

> $M := convert(A, Matrix)$

$$M := \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{9.1.10}$$

> M

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{9.1.11}$$

> $M[i,j]$

Error, bad index into Matrix

Vectors can be represented using the *vector* and *Vector* commands

> $v := vector([v_a, v_b])$

$$v := \begin{bmatrix} v_a & v_b \end{bmatrix} \tag{9.1.12}$$

> v

$$v \tag{9.1.13}$$

> $v[j]$

$$v_j \tag{9.1.14}$$

> $v[2]$

$$v_b \tag{9.1.15}$$

> $V := Vector([V_a, V_b])$

$$V := \begin{bmatrix} V_a \\ V_b \end{bmatrix} \tag{9.1.16}$$

> V

$$\begin{bmatrix} V_a \\ V_b \end{bmatrix} \tag{9.1.17}$$

When using Matrix and Vector, summation and product are performed using `+` and `.`. When using matrix and vector, it is the same but you need to enclose the operation with *evalm*

> $v . A$

$$v \bullet A \tag{9.1.18}$$

> $evalm(\mathbf{(9.1.18)})$

$$\begin{bmatrix} v_a a + v_b c & v_a b + v_b d \end{bmatrix} \tag{9.1.19}$$

Note that for Vector there are row and colum vectors, so

> $V . B$

Error, (in LinearAlgebra:-Multiply) cannot multiply a column Vector and a Matrix

> $V_{row} := Vector[row](V)$

$$V_{row} := \begin{bmatrix} V_a & V_b \end{bmatrix} \tag{9.1.20}$$

> $V_{row} . B$

$$\begin{bmatrix} V_a\, a + V_b\, c & V_a\, b + V_b\, d \end{bmatrix} \tag{9.1.21}$$

The typical operations: conjugate, Transpose, HermitianTranspose, Determinant, Trace, Eigenvalues

> $LinearAlgebra:-Determinant(B)$

$$a\, d - b\, c \tag{9.1.22}$$

> $LinearAlgebra:-Eigenvalues(B)$

$$\begin{bmatrix} \frac{d}{2} + \frac{a}{2} + \frac{\sqrt{a^2 - 2\, a\, d + 4\, b\, c + d^2}}{2} \\ \frac{d}{2} + \frac{a}{2} - \frac{\sqrt{a^2 - 2\, a\, d + 4\, b\, c + d^2}}{2} \end{bmatrix} \tag{9.1.23}$$

> $conjugate(A)$

$$\overline{A} \tag{9.1.24}$$

> $evalm(\%)$

$$\begin{bmatrix} \overline{a} & \overline{b} \\ \overline{c} & \overline{d} \end{bmatrix} \tag{9.1.25}$$

> $LinearAlgebra:-Trace(B)$

$$d + a \tag{9.1.26}$$

Note these do not work with matrix for which you can use the old linalg

> $Trace(A)$

$$Trace(A) \tag{9.1.27}$$

> $linalg[trace](A)$

$$d + a \tag{9.1.28}$$

For solving linear systems there is the LinearSolve command.

Exercises

1. Determine the characteristic matrix, eigenvalues and then: step by step the eigenvectors, of the following matrix:

$$M = \begin{bmatrix} 0 & -\mathrm{I}\sqrt{2} & 0 \\ \mathrm{I}\sqrt{2} & 0 & -\mathrm{I}\sqrt{2} \\ 0 & \mathrm{I}\sqrt{2} & 0 \end{bmatrix}$$

Solution

```
> restart
> with(LinearAlgebra)
```

[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, CompressedSparseForm, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm, FromSplitForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, ProjectionMatrix, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip] **(9.2.1.1)**

```
> M := Matrix(3, (i,j) → if abs(i - j) = 1 then -I sqrt(2) else 0 fi, shape
      = antisymmetric)
```

$$M := \begin{bmatrix} 0 & -I\sqrt{2} & 0 \\ I\sqrt{2} & 0 & -I\sqrt{2} \\ 0 & I\sqrt{2} & 0 \end{bmatrix} \quad \textbf{(9.2.1.2)}$$

```
> CharacteristicMatrix(M, x)
```

$$\begin{bmatrix} x & I\sqrt{2} & 0 \\ -I\sqrt{2} & x & I\sqrt{2} \\ 0 & -I\sqrt{2} & x \end{bmatrix} \quad (9.2.1.3)$$

> *Eigenvalues*(M)

$$\begin{bmatrix} 0 \\ 2 \\ -2 \end{bmatrix} \quad (9.2.1.4)$$

All eigenvectors satisfy

$$M \cdot V = \text{lambda} \cdot V$$

where lambda is an eigenvalue and V is an eigenvector, of the form

> $V := Vector([v_1, v_2, v_3])$

$$V := \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (9.2.1.5)$$

and the v_i, are the unknowns to be determined.

First eigenvector corresponding to the eigenvalue 0

> $M \,.\, V = 0 \cdot V$

$$\begin{bmatrix} -I\sqrt{2}\, v_2 \\ I\sqrt{2}\, v_1 - I\sqrt{2}\, v_3 \\ I\sqrt{2}\, v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (9.2.1.6)$$

> $lhs(\%) - rhs(\%)$

$$\begin{bmatrix} -I\sqrt{2}\, v_2 \\ I\sqrt{2}\, v_1 - I\sqrt{2}\, v_3 \\ I\sqrt{2}\, v_2 \end{bmatrix} \quad (9.2.1.7)$$

> $convert(\%, set)$

$$\left\{-I\sqrt{2}\, v_2, I\sqrt{2}\, v_2, I\sqrt{2}\, v_1 - I\sqrt{2}\, v_3\right\} \quad (9.2.1.8)$$

> $solve(\%)$

$$\{v_1 = v_3, v_2 = 0, v_3 = v_3\} \quad (9.2.1.9)$$

> $V_1 := subs(\%, V)$

(9.2.1.10)

$$V_1 := \begin{bmatrix} v_3 \\ 0 \\ v_3 \end{bmatrix} \tag{9.2.1.10}$$

For the second and third eigenvalues it is the same process, so copy the block of operations above and paste

> $M \,.\, V = 2 \cdot V$

$$\begin{bmatrix} -\mathrm{I}\sqrt{2}\, v_2 \\ \mathrm{I}\sqrt{2}\, v_1 - \mathrm{I}\sqrt{2}\, v_3 \\ \mathrm{I}\sqrt{2}\, v_2 \end{bmatrix} = \begin{bmatrix} 2 v_1 \\ 2 v_2 \\ 2 v_3 \end{bmatrix} \tag{9.2.1.11}$$

> $lhs(\%) - rhs(\%)$

$$\begin{bmatrix} -\mathrm{I}\sqrt{2}\, v_2 - 2 v_1 \\ \mathrm{I}\sqrt{2}\, v_1 - \mathrm{I}\sqrt{2}\, v_3 - 2 v_2 \\ \mathrm{I}\sqrt{2}\, v_2 - 2 v_3 \end{bmatrix} \tag{9.2.1.12}$$

> $convert(\%, set)$

$$\left\{ -\mathrm{I}\sqrt{2}\, v_2 - 2 v_1, \mathrm{I}\sqrt{2}\, v_2 - 2 v_3, \mathrm{I}\sqrt{2}\, v_1 - \mathrm{I}\sqrt{2}\, v_3 - 2 v_2 \right\} \tag{9.2.1.13}$$

> $solve(\%)$

$$\left\{ v_1 = -\frac{\mathrm{I}}{2}\sqrt{2}\, v_2, v_2 = v_2, v_3 = \frac{\mathrm{I}}{2}\sqrt{2}\, v_2 \right\} \tag{9.2.1.14}$$

> $V_2 := subs(\%, V)$

$$V_2 := \begin{bmatrix} -\frac{\mathrm{I}}{2}\sqrt{2}\, v_2 \\ v_2 \\ \frac{\mathrm{I}}{2}\sqrt{2}\, v_2 \end{bmatrix} \tag{9.2.1.15}$$

Now for the third eigenvalue (again copy and paste)

> $M \,.\, V = -2 \cdot V$

$$\begin{bmatrix} -\mathrm{I}\sqrt{2}\, v_2 \\ \mathrm{I}\sqrt{2}\, v_1 - \mathrm{I}\sqrt{2}\, v_3 \\ \mathrm{I}\sqrt{2}\, v_2 \end{bmatrix} = \begin{bmatrix} -2 v_1 \\ -2 v_2 \\ -2 v_3 \end{bmatrix} \tag{9.2.1.16}$$

> $lhs(\%) - rhs(\%)$

$$\begin{bmatrix} -I\sqrt{2}\,v_2 + 2\,v_1 \\ I\sqrt{2}\,v_1 - I\sqrt{2}\,v_3 + 2\,v_2 \\ I\sqrt{2}\,v_2 + 2\,v_3 \end{bmatrix} \quad (9.2.1.17)$$

> convert(%, set)

$$\left\{-I\sqrt{2}\,v_2 + 2\,v_1, I\sqrt{2}\,v_2 + 2\,v_3, I\sqrt{2}\,v_1 - I\sqrt{2}\,v_3 + 2\,v_2\right\} \quad (9.2.1.18)$$

> solve(%)

$$\left\{v_1 = \frac{I}{2}\sqrt{2}\,v_2, v_2 = v_2, v_3 = -\frac{I}{2}\sqrt{2}\,v_2\right\} \quad (9.2.1.19)$$

> $V_3 := subs(\%, V)$

$$V_3 := \begin{bmatrix} \frac{I}{2}\sqrt{2}\,v_2 \\ v_2 \\ -\frac{I}{2}\sqrt{2}\,v_2 \end{bmatrix} \quad (9.2.1.20)$$

So the three eigenvectors are

> V_1, V_2, V_3

$$\begin{bmatrix} v_3 \\ 0 \\ v_3 \end{bmatrix}, \begin{bmatrix} -\frac{I}{2}\sqrt{2}\,v_2 \\ v_2 \\ \frac{I}{2}\sqrt{2}\,v_2 \end{bmatrix}, \begin{bmatrix} \frac{I}{2}\sqrt{2}\,v_2 \\ v_2 \\ -\frac{I}{2}\sqrt{2}\,v_2 \end{bmatrix} \quad (9.2.1.21)$$

>

6. Vector Analysis

Algebraic representation	Any symbol (could contain many letters) that ends with the underscore _ . It can have indices and functionality, as in A_[x](t), displayed as $\vec{A}_x(t)$. You can compute with vectors without projecting them into any orthonormal basis. Unit vectors for Cartesian, cylindrical and spherical coordinates: use the standard names preceded by the underscore, as in [_i, _j, _k] displayed as $[\hat{i}, \hat{j}, \hat{k}]$ You can represent a projected vector as an algebraic sum of products of components times unit vectors.
Operations	+ . &x, respectively for plus, scalar product and cross product
Command	Those of the Physics:-Vectors package,

s	[*&x*, `+`, `.`, *ChangeBasis*, *Component*, *Curl*, *DirectionalDiff*, *Divergence*, *Gradient*, *Identify*, *Laplacian*,∇, *Norm*, *Setup*, *diff*] and their inert forms, prefixing the command with %, as in %Gradient.
Projected vectors using a matricial representation	See the VectorCalculus package.

Table 6: Vector Analysis

Examples

Vectorial equation of a plane

Problem

1. Derive the *vectorial* equation of a plane passing through three generic points **A**, **B**, and **C**.
2. Choose three concrete points **A**, **B**, and **C** and plot this plane.

Solution

The *vectorial equation of a plane* is the equation satisfied by the position vector of *any* point of this plane.

```
> restart;
  with(Physics[Vectors]) :
  Setup(mathematicalnotation = true);
```

$$[mathematicalnotation = true] \qquad \textbf{(10.1.1.1)}$$

1. To construct this equation, let $\vec{A}$, $\vec{B}$, and $\vec{C}$ be the vectors pointing to **A, B**, and **C**, respectively, and $\vec{r}$ be the position vector of *any* point of this plane. The differences $\vec{A} - \vec{C}$ and $\vec{A} - \vec{B}$ are vectors parallel to the plane we want to represent, and also the difference between $\vec{r}$ and any of $\vec{A}$, $\vec{B}$, or $\vec{C}$ is a vector parallel to the plane. So the equation of the plane can be obtained by taking the cross product of differences involving $\vec{A}$, $\vec{B}$, and $\vec{C}$ to construct a vector *perpendicular* to the plane,

```
> G_ := (A_ - B_) &x (A_ - C_);
```

$$\vec{G} := \left(\vec{A} - \vec{B}\right) \times \left(\vec{A} - \vec{C}\right) \qquad \textbf{(10.1.1.2)}$$

then equating to zero the scalar product of $\vec{G}$ with any of the differences parallel to the plane involving $\vec{r}$. So for instance, one way of writing this vectorial equation of the plane is

```
> Eq := (r_ - A_) . G_ = 0;
```

$$Eq := \left(\vec{r} - \vec{A}\right) \cdot \left(\left(\vec{A} - \vec{B}\right) \times \left(\vec{A} - \vec{C}\right)\right) = 0 \qquad \textbf{(10.1.1.3)}$$

2. To plot this plane, turn the generic points **A**, **B**, and **C** into concrete points; that is, give values to the components of $\vec{A}$, $\vec{B}$, and $\vec{C}$. For example:

```
> A_ := 2 _i - 3 _j + 4 _k
```

$$\vec{A} := 2\,\hat{i} - 3\,\hat{j} + 4\,\hat{k} \qquad \textbf{(10.1.1.4)}$$

> $B\_ := 5\,\_i + 4\,\_j - 7\,\_k;$

$$\vec{B} := 5\,\hat{i} + 4\,\hat{j} - 7\,\hat{k} \qquad \textbf{(10.1.1.5)}$$

> $C\_ := \frac{30}{4}\,A\_ + \frac{90}{7}\,B\_;$

$$\vec{C} := \frac{555\,\hat{i}}{7} + \frac{405\,\hat{j}}{14} - 60\,\hat{k} \qquad \textbf{(10.1.1.6)}$$

For $\vec{r}$ we always have

> $r\_ := x\,\_i + y\,\_j + z\,\_k;$

$$\vec{r} := x\,\hat{i} + y\,\hat{j} + z\,\hat{k} \qquad \textbf{(10.1.1.7)}$$

The vectorial equation for these particular points **A**, **B**, and **C** is thus:

> $Eq;$

$$-\frac{1355\,x}{14} - \frac{4607\,y}{7} - \frac{6233\,z}{14} = 0 \qquad \textbf{(10.1.1.8)}$$

As a verification that the surface represented by this equation contains the points **A**, **B**, and **C**, you can substitute in the values of the coordinates of these points and see that the equation is satisfied. These are the coordinates of the three points:

> $A, B, C := seq([seq(Component(v,j), j = 1..3)], v = [A\_, B\_, C\_]);$

$$A, B, C := [2, -3, 4], [5, 4, -7], \left[\frac{555}{7}, \frac{405}{14}, -60\right] \qquad \textbf{(10.1.1.9)}$$

> **for** P **in** $[A, B, C]$ **do**
> $eval(Eq, [x = P[1], y = P[2], z = P[3]]);$
> **od**;

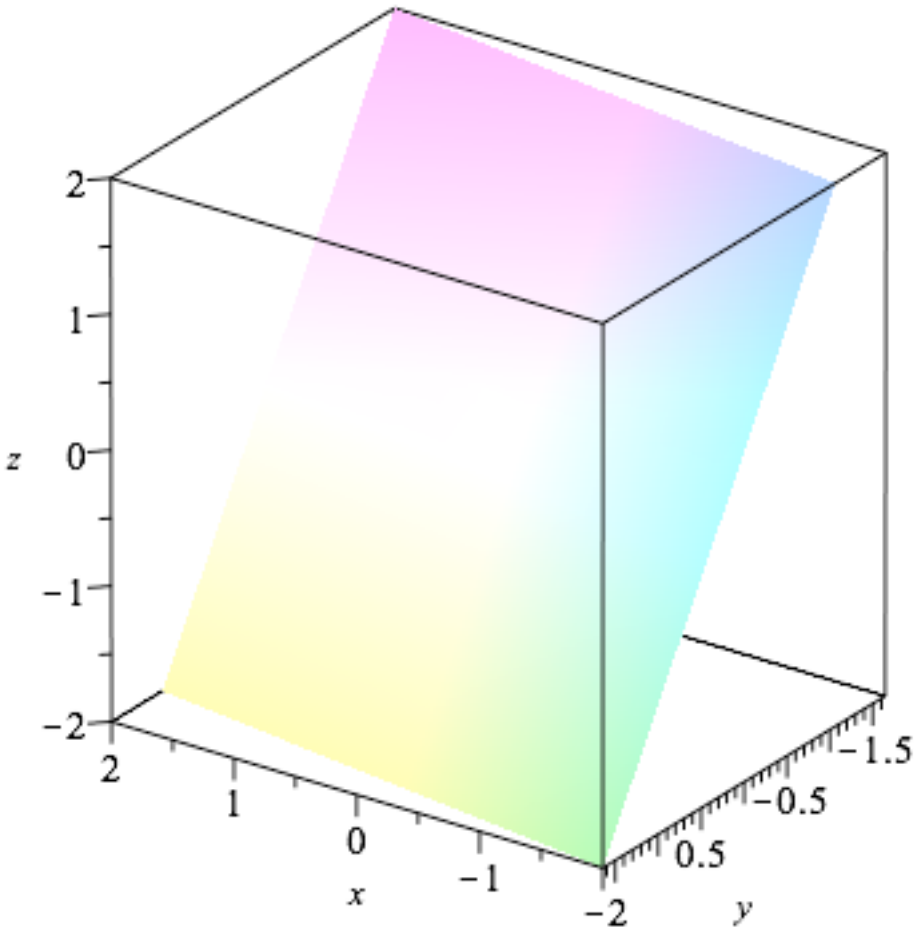
$$0 = 0$$
$$0 = 0$$
$$0 = 0 \qquad \textbf{(10.1.1.10)}$$

That this surface is a plane is clear from the fact that *Eq* is linear in all of *x*, *y*, and *z*. One way of plotting this plane is to use the command implicitplot3d.

> $opts := axes = boxed, scaling = constrained, orientation = [125, 65], style = surface;$

$$opts := axes = boxed, scaling = constrained, orientation = [125, 65], style = surface \qquad \textbf{(10.1.1.11)}$$

> $plots[implicitplot3d](Eq, x = -2..2, y = -2..2, z = -2..2, opts);$



Volume element of a sphere

Problem
Determine the infinitesimal volume element of a sphere expressed in spherical coordinates.

Solution
Let $\vec{r} = \vec{R}(u, v, w)$ be the vectorial equation, parameterized by u, v, and w, of a generic 3-D geometric object; in this case, we are dealing with a sphere of generic radius r. The volume element is derived from equation as $d^3\vec{r} = \frac{\partial}{\partial u}\vec{r} \cdot \left(\frac{\partial}{\partial v}\vec{r} \times \frac{\partial}{\partial w}\vec{r} \right) du\, dv\, dw$.

```
> restart;
  with(Physics[Vectors]) :
  Setup(mathematicalnotation = true);
```

[*mathematicalnotation = true*] **(10.1.2.1)**

We want this volume element expressed in spherical coordinates (*r*, phi, theta); we can always choose these coordinates themselves as parameters *u, v, w*. We are thus interested in the explicit form of (note the use of **%diff**, the inert form diff):

```
> answer := %diff(r_, r) . (%diff(r_, theta) &x %diff(r_, phi));
```

$$answer := \frac{\partial}{\partial r}\vec{r} \cdot \left(\frac{\partial}{\partial \theta}\vec{r} \times \frac{\partial}{\partial \phi}\vec{r} \right) \qquad \textbf{(10.1.2.2)}$$

The first step is to write the vectorial equation $\vec{r} = \vec{R}(r, \text{phi}, \text{theta})$ for a sphere of radius r; that is, the equation satisfied by the position vector $\vec{r}$ of *any* point of the sphere. In spherical coordinates, and choosing the origin of the reference system at the center of the sphere, this vectorial equation has its simplest form $\vec{R}(r, \text{phi}, \text{theta}) = r\,\hat{r}$, where $\vec{r}$ points to *any* point of the sphere, r is the radial coordinate (constant over the sphere) and $\hat{r}$ is the radial unit vector. So

```
> r_ := r _r
```

$$\vec{r} := r\,\hat{r} \qquad \textbf{(10.1.2.3)}$$

From where the value of

```
> answer;
```

$$\frac{\partial}{\partial r}(r\,\hat{r}) \cdot \left(\frac{\partial}{\partial \theta}(r\,\hat{r}) \times \frac{\partial}{\partial \phi}(r\,\hat{r}) \right) \qquad \textbf{(10.1.2.4)}$$

can be computed directly, using the value command

```
> value(answer);
```

$$r^2 \sin(\theta) \qquad \textbf{(10.1.2.5)}$$

Alternatively, one could compute this result one step at a time, making explicit that $\hat{r}$ depends on φ and θ. For that purpose we change the basis in the vectorial equation to the Cartesian basis $(\hat{i}, \hat{j}, \hat{k})$, where all the unit vectors are constant and so the partial derivatives can be performed directly.

```
> r_ := ChangeBasis(r_, 1);
```

$$\vec{r} := r \sin(\theta) \cos(\phi)\,\hat{i} + r \sin(\theta) \sin(\phi)\,\hat{j} + r \cos(\theta)\,\hat{k} \qquad \textbf{(10.1.2.6)}$$

So, the *answer* introduced lines above becomes:

```
> answer;
```

$$\frac{\partial}{\partial r}\left(r \sin(\theta) \cos(\phi)\,\hat{i} + r \sin(\theta) \sin(\phi)\,\hat{j} + r \cos(\theta)\,\hat{k}\right) \cdot \left(\frac{\partial}{\partial \theta}\left(r \sin(\theta) \cos(\phi)\,\hat{i} + r \sin(\theta) \sin(\phi)\,\hat{j} + r \cos(\theta)\,\hat{k}\right) \times \frac{\partial}{\partial \phi}\left(r \sin(\theta) \cos(\phi)\,\hat{i} + r \sin(\theta) \sin(\phi)\,\hat{j} + r \cos(\theta)\,\hat{k}\right) \right) \qquad \textbf{(10.1.2.7)}$$

```
> value((10.1.2.7))
```

$$r^2 \sin(\theta) \qquad \textbf{(10.1.2.8)}$$

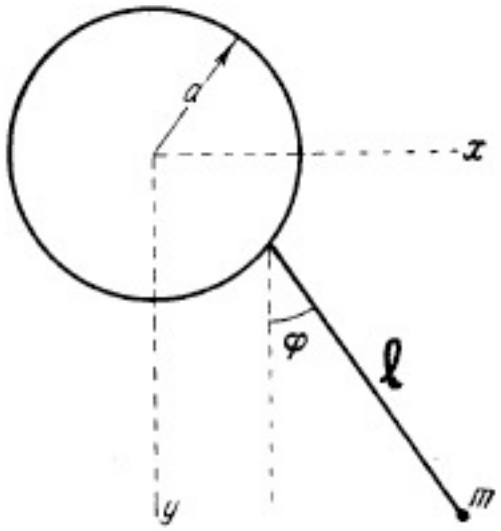
Hence the volume of element requested is $d^3\vec{r} = \sin(\text{theta})\, r^2\, d\,r\, d\,\theta\, d\,\varphi$.

Lagrangian for a pendulum

Problem

Determine the Lagrangian of a plane pendulum having a mass m in its extremity and whose suspension point:

a) moves uniformly over a vertical circumference with a constant frequency ω.



b) oscillates horizontally on the plane of the pendulum according to $x = \cos(\text{omega}\ t)$.

Solution

The Lagrangian is defined as

```
> restart;
  with(Physics[Vectors]);
  Setup(mathematicalnotation = true)
```

$$[\&x, `+`, `.`, ChangeBasis, Component, Curl, DirectionalDiff, Divergence, Gradient, Identify, Laplacian, \nabla, Norm, Setup, diff]$$

$$[mathematicalnotation = true] \qquad \textbf{(10.1.3.1.1)}$$

```
> L := T - U
```

$$L := T - U \qquad \textbf{(10.1.3.1.2)}$$

where T and U are the kinetic and potential energy of the system, respectively, in this case constituted by a single point of mass m. The potential energy U is the gravitational energy

```
> U := -m g y
```

$$U := -m\,g\,y \qquad \textbf{(10.1.3.1.3)}$$

where g is the gravitational constant and we choose the y axis along the vertical, pointing downwards, so that the gravitational force $\vec{F}_g = mg\,\hat{j}$. The kinetic energy is:

```
> T := 1/2 m v_ . v_;
```

$$T := \frac{m\,\|\vec{v}\|^2}{2} \qquad \textbf{(10.1.3.1.4)}$$

To compute this velocity, the position vector $\vec{r}$ of the suspension point of the pendulum,

```
> r_ := x _i + y _j;
```

$$\vec{r} := x\,\hat{i} + y\,\hat{j} \qquad \textbf{(10.1.3.1.5)}$$

must be determined. Choosing the x axis horizontally and the origin of the reference system

at the center of the circle (see figure above), the x and y coordinates are given by:

```
> parametric_equations := [x = a cos(omega t) + l sin(phi(t)), y =
      -a sin(omega t) + l cos(phi(t))];
```

$$parametric\_equations := [x = a\cos(\omega t) + l\sin(\phi(t)), y = -a\sin(\omega t) + l\cos(\phi(t))] \quad \textbf{(10.1.3.1.6)}$$

```
> r_ := eval(r_, parametric_equations);
```

$$\vec{r} := (a\cos(\omega t) + l\sin(\phi(t)))\,\hat{i} + (-a\sin(\omega t) + l\cos(\phi(t)))\,\hat{j} \quad \textbf{(10.1.3.1.7)}$$

```
> v_ := diff(r_, t);
```

$$\vec{v} := (-a\omega\sin(\omega t) + l\dot{\phi}(t)\cos(\phi(t)))\,\hat{i} + (-a\omega\cos(\omega t) - l\dot{\phi}(t)\sin(\phi(t)))\,\hat{j} \quad \textbf{(10.1.3.1.8)}$$

```
> T;
```

$$\frac{1}{2}\left(m\left((-a\omega\sin(\omega t) + l\dot{\phi}(t)\cos(\phi(t)))^2 + (-a\omega\cos(\omega t) - l\dot{\phi}(t)\sin(\phi(t)))^2\right)\right) \quad \textbf{(10.1.3.1.9)}$$

This expression contains products of trigonometric functions, so one simplification consists of combining these products.

```
> combine(T, trig)
```

$$-\dot{\phi}(t)\sin(\omega t - \phi(t))\,a\,l\,m\,\omega + \frac{\dot{\phi}(t)^2 l^2 m}{2} + \frac{a^2 m \omega^2}{2} \quad \textbf{(10.1.3.1.10)}$$

For the gravitational energy, expressed in terms of the parametric equations of the point of mass m, we have

```
> U := eval(U, parametric_equations)
```

$$U := -m\,g\,(-a\sin(\omega t) + l\cos(\phi(t))) \quad \textbf{(10.1.3.1.11)}$$

So the requested Lagrangian is

```
> L := combine(L, trig)
```

$$L := -\dot{\phi}(t)\sin(\omega t - \phi(t))\,a\,l\,m\,\omega + \frac{\dot{\phi}(t)^2 l^2 m}{2} + \frac{a^2 m \omega^2}{2} - \sin(\omega t)\,a\,g\,m + \cos(\phi(t))\,g\,l\,m \quad \textbf{(10.1.3.1.12)}$$

Taking into account that the Lagrangian of a system is defined up to a total derivative with respect to t, we can eliminate the two terms that can be rewritten as total derivatives; these are $\frac{m\,a^2\,\mathrm{omega}^2}{2}$ and $m\,g\,a\,\sin(\mathrm{omega}\,t)$, so

```
> ω^2
```

$$\omega^2 \quad \textbf{(10.1.3.1.13)}$$

```
> select(has, L, [omega^2, sin(omega t)])
```

$$\frac{a^2 m \omega^2}{2} - \sin(\omega t)\,a\,g\,m \quad \textbf{(10.1.3.1.14)}$$

```
> L := L - (10.1.3.1.14)
```

$$L := -\dot{\mathrm{phi}}(t)\sin(\omega t - \phi(t))\, a\, l\, m\, \omega + \frac{\dot{\mathrm{phi}}(t)^2\, l^2\, m}{2} + \cos(\phi(t))\, g\, l\, m \quad \textbf{(10.1.3.1.15)}$$

---

b) The steps are the same as in part a:

```
> restart;
  with(Physics[Vectors]):
  Setup(mathematicalnotation = true)
```

$$[mathematicalnotation = true] \quad \textbf{(10.1.3.1.16)}$$

```
> L := T - U
```

$$L := T - U \quad \textbf{(10.1.3.1.17)}$$

```
> U := -m g y
```

$$U := -m\, g\, y \quad \textbf{(10.1.3.1.18)}$$

```
> T := 1/2 m v_ . v_
```

$$T := \frac{m\, \|\vec{v}\|^2}{2} \quad \textbf{(10.1.3.1.19)}$$

```
> r_ := x _i + y _j;
```

$$\vec{r} := x\,\hat{i} + y\,\hat{j} \quad \textbf{(10.1.3.1.20)}$$

Now, regarding part a), the only change is in the expression of the y coordinate, which for this part b) is:

```
> y = l cos(phi(t));
```

$$y = l\cos(\phi(t)) \quad \textbf{(10.1.3.1.21)}$$

So the parametric equations in this case are

```
> parametric_equations := [x = a cos(omega t) + (l sin(phi(t))), (10.1.3.1.21)]
```

$$parametric\_equations := [x = a\cos(\omega t) + l\sin(\phi(t)), y = l\cos(\phi(t))] \quad \textbf{(10.1.3.1.22)}$$

```
> r := eval(r, parametric_equations)
```

$$\vec{r} := (a\cos(\omega t) + l\sin(\phi(t)))\,\hat{i} + l\cos(\phi(t))\,\hat{j} \quad \textbf{(10.1.3.1.23)}$$

```
> v := diff(r, t)
```

$$\vec{v} := (-a\,\omega\sin(\omega t) + l\,\dot{\mathrm{phi}}(t)\cos(\phi(t)))\,\hat{i} - l\,\dot{\mathrm{phi}}(t)\sin(\phi(t))\,\hat{j} \quad \textbf{(10.1.3.1.24)}$$

For the gravitational energy, expressed in terms of the parametric equations of the point of mass m, we have

```
> U := eval(U, parametric_equations)
```

$$U := -m\, g\, l\cos(\phi(t)) \quad \textbf{(10.1.3.1.25)}$$

So the requested Lagrangian is

```
> L
```

$$\frac{m\left((-a\,\omega\sin(\omega t) + l\,\dot{\mathrm{phi}}(t)\cos(\phi(t)))^2 + l^2\,\dot{\mathrm{phi}}(t)^2\sin(\phi(t))^2\right)}{2} \quad \textbf{(10.1.3.1.26)}$$

$$+ m\,g\,l\cos(\phi(t))$$

```
> L := combine(L, trig)
```

$$L := -\frac{\dot{\phi}(t)\sin(\omega t+\phi(t))\,a\,l\,m\,\omega}{2} - \frac{\dot{\phi}(t)\sin(\omega t-\phi(t))\,a\,l\,m\,\omega}{2} - \frac{\cos(2\,\omega t)\,a^2\,m\,\omega^2}{4} + \frac{\dot{\phi}(t)^2\,l^2\,m}{2} + \frac{a^2\,m\,\omega^2}{4} + m\,g\,l\cos(\phi(t)) \quad \textbf{(10.1.3.1.27)}$$

The terms in L that can be expressed as total derivatives can be discarded, so

```
> select(has, L, [omega^2, cos(2 omega t)])
```

$$-\frac{\cos(2\,\omega t)\,a^2\,m\,\omega^2}{4} + \frac{a^2\,m\,\omega^2}{4} \quad \textbf{(10.1.3.1.28)}$$

So the Lagrangian is

```
> L := L - (10.1.3.1.28)
```

$$L := -\frac{\dot{\phi}(t)\sin(\omega t+\phi(t))\,a\,l\,m\,\omega}{2} - \frac{\dot{\phi}(t)\sin(\omega t-\phi(t))\,a\,l\,m\,\omega}{2} + \frac{\dot{\phi}(t)^2\,l^2\,m}{2} + m\,g\,l\cos(\phi(t)) \quad \textbf{(10.1.3.1.29)}$$

Exercises

Vectorial equation of a plane tangent to a sphere of radius a

Problem

Derive the *vectorial* equation of a plane tangent to a sphere of radius a.

Solution

The *vectorial equation* of this plane is the equation satisfied by the position vector of *any* point of it.

```
> restart;
  with(Physics[Vectors]):
  Setup(mathematicalnotation = true)
```

$$[mathematicalnotation = true] \quad \textbf{(10.2.1.1.1)}$$

Let $\vec{r}$ represent the position vector of *any* point of the plane, $\vec{A}$ be a vector pointing to the center of the sphere, and $\vec{B}$ be a vector pointing to the point **B** where the plane is tangent to the sphere. So the difference $\vec{r} - \vec{B}$ is a vector on the plane, and the difference $\vec{B} - \vec{A}$ is a vector from the center of the sphere to the point of contact between the sphere and the tangent plane (that is, a vector perpendicular to the plane). Hence, these two vectors are perpendicular, and so their scalar product is equal to zero.

```
> Eq := (r_ - B_) . (B_ - A_) = 0
```

(10.2.1.1.2)

$$Eq := (\vec{r} - \vec{B}) \cdot (\vec{B} - \vec{A}) = 0 \qquad \textbf{(10.2.1.1.2)}$$

This is already the vectorial equation of the tangent plane but not yet expressed in terms of the radius of the sphere. Now since $\vec{B} - \vec{A}$ is a vector from the center of the sphere to the point of contact with the plane, the norm of this vector is the radius a of the sphere.

> $key := Norm(B\_ - A\_) = a$

$$key := \left\| \vec{B} - \vec{A} \right\| = a \qquad \textbf{(10.2.1.1.3)}$$

Expand both expressions to use them together.

> $expand(Eq);$

$$\vec{r} \cdot \vec{B} - \vec{r} \cdot \vec{A} - \left\| \vec{B} \right\|^2 + \vec{B} \cdot \vec{A} = 0 \qquad \textbf{(10.2.1.1.4)}$$

> $map(u \rightarrow u\text{^}2, key);$

$$\left\| \vec{B} - \vec{A} \right\|^2 = a^2 \qquad \textbf{(10.2.1.1.5)}$$

> $expand(\textbf{(10.2.1.1.5)})$

$$\left\| \vec{B} \right\|^2 - 2\,\vec{B} \cdot \vec{A} + \left\| \vec{A} \right\|^2 = a^2 \qquad \textbf{(10.2.1.1.6)}$$

So simplify one with respect to the other one, eliminating $\vec{B}$ &. $\vec{B}$ (see simplify/siderels)

> $simplify(\textbf{(10.2.1.1.4)}, \{expand(\textbf{(10.2.1.1.5)})\}, \{Norm(B\_)\})$

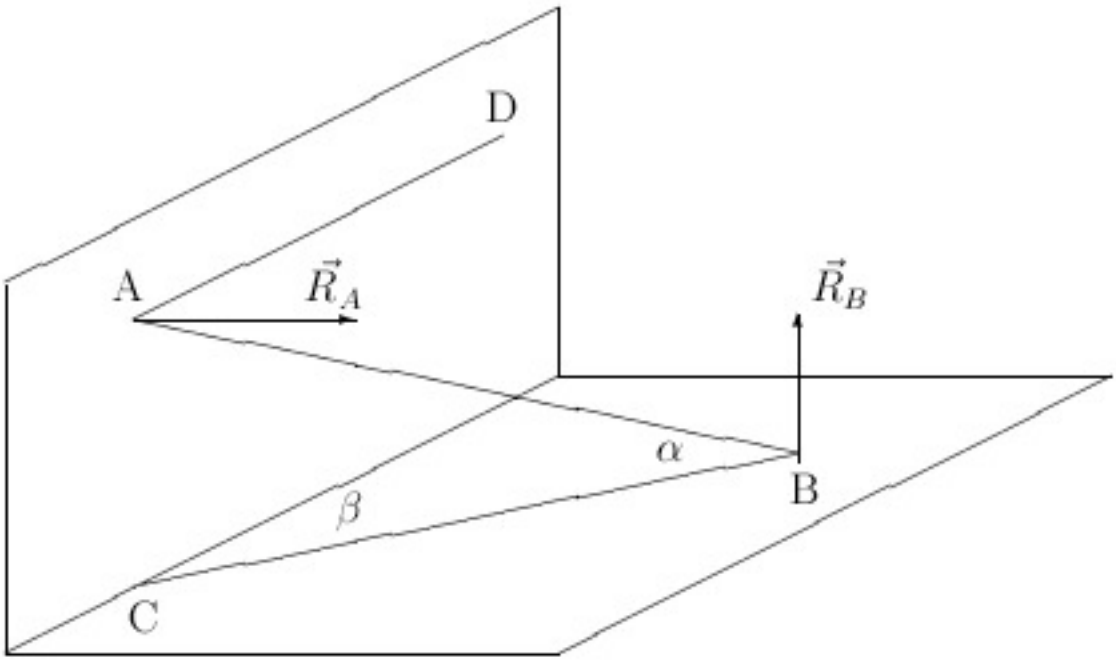
$$\left\| \vec{A} \right\|^2 - a^2 - \vec{B} \cdot \vec{A} + \vec{r} \cdot \vec{B} - \vec{r} \cdot \vec{A} = 0 \qquad \textbf{(10.2.1.1.7)}$$

Collecting terms in the above, this vectorial equation requested can be rewritten more compactly as $(\vec{B} - \vec{A}) \cdot (\vec{r} - \vec{A}) = a^2$. As an exercise, consider choosing three concrete values for the positions of **A**, **B**, and the radius a of the sphere, then insert these values into the equation derived, and plot the sphere and this tangent plane together (see plots[display] to merge the plots).

Static: reactions of planes and tensions on cables

Problem

A bar **AB** of weight **w** and length **L** has one extreme on a horizontal plane and the other on a vertical place, and is kept in that position by two cables **AD** and **BC**. The bar forms an angle alpha with the horizontal plane and its projection **BC** over this plane forms an angle beta with the vertical plane. The cable **BC** is in the vertical plane where the bar is. Determine the reactions of the planes at **A** and **B** and the tensions on the cables.



Solution

There are two equations that contain information about the state of equilibrium of a system. The first one, saying that the sum of the forces acting on the body are equal to zero, tells that the center of mass of the body is not accelerated. The second one, saying that the sum of the moments of the forces acting on the body (that is, the total torque) is zero tells that the rotation of the body around its center of masses is not changing (if it is not rotating, it stays that way). These two equations involve the reactions of the planes and the tensions on the cables, so from them we can obtain the solution to the problem. There is no friction so it is also clear that the reactions $\vec{R}_A$ and $\vec{R}_B$ are perpendicular to the planes, as shown in the figure, and the tensions $\vec{T}_A$ and $\vec{T}_B$ on the cables have direction **AD** and **BC**, respectively.

The steps to solve this problem are:

1. Determine each force $\vec{F}$ acting on the bar and its application point $\vec{r}$.
2. Equate the sum of the forces $\vec{F}$ to zero.
3. Equate the sum of the moments $\vec{r} \times \vec{F}$ to zero.
4. Solve these two vectorial equations for $\vec{R}_A$, $\vec{R}_B$, $\vec{T}_A$, and $\vec{T}_B$, representing the reactions of the planes at the points of contact **A** and **B**, and the tensions of the cables attached to the bar at **A** and **B**, respectively.

```
> restart;
  with(Physics[Vectors]);
  Setup(mathematicalnotation = true)
[&x, `+`, `.`, ChangeBasis, Component, Curl, DirectionalDiff, Divergence,
    Gradient, Identify, Laplacian, ∇, Norm, Setup, diff]
                    [mathematicalnotation = true]                    (10.2.2.1.1)
```

The forces acting on the bar are its weight $\vec{w}$, and the reactions and tensions $\vec{R}_A$, $\vec{R}_B$, $\vec{T}_A$, and $\vec{T}_B$. So the two equilibrium equations are:

```
> eq[1] := w_ + R_[A] + R_[B] + T_[A] + T_[B] = 0;
```

$$eq_1 := \vec{w} + \vec{R}_A + \vec{R}_B + \vec{T}_A + \vec{T}_B = 0 \qquad \textbf{(10.2.2.1.2)}$$

```
> eq[2] := r_[w] &x w_ + r_[A] &x R_[A] + r_[B] &x R_[B] + r_[A]
      &x T_[A] + r_[B] &x T_[B] = 0;
```

$$eq_2 := \vec{r}_w \times \vec{w} + \vec{r}_A \times \vec{R}_A + \vec{r}_B \times \vec{R}_B + \vec{r}_A \times \vec{T}_A + \vec{r}_B \times \vec{T}_B = 0 \qquad \textbf{(10.2.2.1.3)}$$

Set the origin and orientation of the reference system to project these vectors; any choice will do, but a good one will simplify the algebraic manipulations. We choose the origin at the point **B**, the vertical z axis in the direction of the reaction $\vec{R}_B$, so that $\vec{r}_B = 0$, the y axis in the direction of $\vec{R}_A$, and the x axis in the remaining direction, anti-parallel to $\vec{T}_A$. With this choice, the vectors entering eq_1 and eq_2 are projected as follows:

```
> R_[B] := abs(R[B]) * _k;
```

$$\vec{R}_B := |R_B|\,\hat{k} \qquad \textbf{(10.2.2.1.4)}$$

where $|R_B|$ represents the norm of $\vec{R}_B$, to be determined,

```
> r_[B] := 0;
```

$$\vec{r}_B := 0 \qquad \textbf{(10.2.2.1.5)}$$

```
> R_[A] := abs(R[A]) * _j;
```

$$\vec{R}_A := |R_A|\,\hat{j} \qquad \textbf{(10.2.2.1.6)}$$

where $|R_A|$ is to be determined. This reaction $\vec{R}_A$ is applied to the bar at **A,** represented by $\vec{r}_A$; its component along the x axis is obtained by projecting the segment **BA** onto the horizontal plane ($L\cos(\text{alpha})$), resulting in **BC**, then into the intersection of the two planes. So:

```
> r_[A] := L * cos(alpha) * (cos(beta) * _i + sin(2 * Pi - beta) * _j) + L
      * sin(alpha) * _k;
```

$$\vec{r}_A := L\cos(\beta)\cos(\alpha)\,\hat{i} - L\sin(\beta)\cos(\alpha)\,\hat{j} + L\sin(\alpha)\,\hat{k} \qquad \textbf{(10.2.2.1.7)}$$

For the other vectors we have

```
> T_[A] := - abs(T[A]) * _i;
```

$$\vec{T}_A := -|T_A|\,\hat{i} \qquad \textbf{(10.2.2.1.8)}$$

```
> T_[B] := abs(T[B]) * cos(beta) * _i + abs(T[B]) * sin(2 * Pi - beta) * _j;
```

$$\vec{T}_B := |T_B|\cos(\beta)\,\hat{i} - |T_B|\sin(\beta)\,\hat{j} \qquad \textbf{(10.2.2.1.9)}$$

where $|T_A|$ and $|T_B|$ are to be determined,

```
> w_ := -abs(w) * _k;
```

$$\vec{w} := -|w|\,\hat{k} \qquad \textbf{(10.2.2.1.10)}$$

```
> r_[w] := r_[A]/2;
```

$$\vec{r}_w := \frac{L\cos(\beta)\cos(\alpha)\,\hat{i}}{2} - \frac{L\sin(\beta)\cos(\alpha)\,\hat{j}}{2} + \frac{L\sin(\alpha)\,\hat{k}}{2} \quad \textbf{(10.2.2.1.11)}$$

The two equilibrium equations now appear as

> $eq[1];$

$$-|w|\,\hat{k} + |R_A|\,\hat{j} + |R_B|\,\hat{k} - |T_A|\,\hat{i} + |T_B|\cos(\beta)\,\hat{i} - |T_B|\sin(\beta)\,\hat{j} = 0 \quad \textbf{(10.2.2.1.12)}$$

> $eq[2];$

$$\frac{L\sin(\beta)\cos(\alpha)\,|w|\,\hat{i}}{2} + \frac{L\cos(\beta)\cos(\alpha)\,|w|\,\hat{j}}{2} - L\sin(\alpha)\,|R_A|\,\hat{i} + L\cos(\beta)\cos(\alpha)\,|R_A|\,\hat{k} - L\sin(\alpha)\,|T_A|\,\hat{j} - L\sin(\beta)\cos(\alpha)\,|T_A|\,\hat{k} = 0 \quad \textbf{(10.2.2.1.13)}$$

These two vectorial equations represent a system of six equations, obtained by equating each of the coefficients of $\hat{i}$, $\hat{j}$, and $\hat{k}$ in each of the equations to zero; that is, taking the components of the vectorial equations along each axis:

> $Eq[1,2,3] := seq(Component(lhs(eq[1]),n) = 0, n = 1..3);$

$$Eq_{1,2,3} := -|T_A| + |T_B|\cos(\beta) = 0, |R_A| - |T_B|\sin(\beta) = 0, -|w| + |R_B| = 0 \quad \textbf{(10.2.2.1.14)}$$

> $Eq[4,5,6] := seq(Component(lhs(eq[2]),n) = 0, n = 1..3);$

$$Eq_{4,5,6} := \frac{L\sin(\beta)\cos(\alpha)\,|w|}{2} - L\sin(\alpha)\,|R_A| = 0, \frac{L\cos(\beta)\cos(\alpha)\,|w|}{2} - L\sin(\alpha)\,|T_A| = 0, L\cos(\beta)\cos(\alpha)\,|R_A| - L\sin(\beta)\cos(\alpha)\,|T_A| = 0 \quad \textbf{(10.2.2.1.15)}$$

So the system of equations to be solved is

> $sys := \{Eq[1,2,3], Eq[4,5,6]\}:$

The unknowns are

> $var := \{abs(R[A]), abs(R[B]), abs(T[A]), abs(T[B])\};$

$$var := \{|R_A|, |R_B|, |T_A|, |T_B|\} \quad \textbf{(10.2.2.1.16)}$$

and the solution is

> $solve(sys, var);$

$$\left\{|R_A| = \frac{\cos(\alpha)\,|w|\sin(\beta)}{2\sin(\alpha)}, |R_B| = |w|, |T_A| = \frac{\cos(\alpha)\,|w|\cos(\beta)}{2\sin(\alpha)}, |T_B| = \frac{\cos(\alpha)\,|w|}{2\sin(\alpha)}\right\} \quad \textbf{(10.2.2.1.17)}$$

Potential Phi *and electric field* $\vec{E}$ *of a charged disk*

Problem

Calculate the potential Phi and the electric field $\vec{E}$ of a disc of radius *a,* loaded with a surface

density of charge sigma (constant), in points of an axis perpendicular to the disk and passing through its center.

Solution

Given the potential Phi, the general expression of the electric field $\vec{E}$ is

$$\vec{E} = -\nabla(\mathrm{Phi})$$

In turn, the general expression for the potential Phi of a distribution of charges over a surface is given by

$$\mathrm{Phi}(\vec{r}) = \int_{\mathrm{Sigma}} \frac{\mathrm{sigma}(\vec{R})}{|\vec{R} - \vec{r}|}\, ds$$

where $\vec{r}$ is the position vector of any point in space, $\vec{R}$ is the position vector of any point of the disk, and *ds* is the surface element; the above expression is a surface integral with Sigma representing the integration domain.

```
> restart;
  with(Physics[Vectors]);
  Setup(mathematicalnotation = true)
```

```
[&x, `+`, `.`, ChangeBasis, Component, Curl, DirectionalDiff, Divergence,
   Gradient, Identify, Laplacian, ∇, Norm, Setup, diff]
                    [mathematicalnotation = true]
```

(10.2.3.1.1)

The expression for the electric field $\vec{E}$ as the gradient of the potential Φ can be entered as

```
> E_ := -%Gradient(Phi);
```

$$\vec{E} := -\nabla\Phi$$

(10.2.3.1.2)

where in the above we are using the *inert* form of the Gradient command. The expression for Phi in turn can be entered as a double integral in cylindrical coordinates (rho, phi, *z*); the element of surface of a disk in these coordinates is $\rho\, d\rho\, d\varphi$, with rho varying from 0 to *a* and phi from 0 to 2 pi .

```
> Phi := Int(Int(sigma * rho / Norm(r_ - R_), rho = 0..a), phi = 0..2 * Pi);
```

$$\Phi := \int_0^{2\pi}\int_0^{a} \frac{\sigma\,\rho}{\|\vec{r} - \vec{R}\|}\, d\rho\, d\phi$$

(10.2.3.1.3)

We choose the origin of the system of references to be the center of the disk and the *z* axis oriented perpendicular to the disk. In this system of references, the position vector of a point over the *z* axis is

```
> r_ := z * _k;
```

$$\vec{r} := z\,\hat{k}$$

(10.2.3.1.4)

and the position vector of a point of the disk is

```
> R_ := rho * _rho;
```

$$\vec{R} := \rho\,\hat{\rho}$$

(10.2.3.1.5)

The value of the potential Phi for $z > 0$ can now be computed.

```
> Phi := value(Phi) assuming z > 0;
```

$$\Phi := -2\,z\,\sigma\,\pi + 2\sqrt{a^2+z^2}\;\sigma\,\pi \qquad \textbf{(10.2.3.1.6)}$$

From this we get

```
> E_;
```

$$-\nabla\left(-2\,z\,\sigma\,\pi + 2\sqrt{a^2+z^2}\;\sigma\,\pi\right) \qquad \textbf{(10.2.3.1.7)}$$

```
> E_ := value((10.2.3.1.7));
```

$$\vec{E} := \frac{2\,\pi\,\sigma\left(\sqrt{a^2+z^2}-z\right)\hat{k}}{\sqrt{a^2+z^2}} \qquad \textbf{(10.2.3.1.8)}$$

Magnetic field $\vec{H}$ of a rotating charged disk

Problem

A disk of radius a, uniformly charged with a surface density of charge σ, rotates around its axis with a constant angular velocity phi = ω, where phi is the cylindrical coordinate (the polar angle). Calculate the magnetic field on the axis of the disk.

Solution

The expression of the magnetic field $\vec{H}(\vec{r})$ due to a current of charges $\vec{J}(\vec{R})$ is

$$\vec{H}(\vec{r}) = \int_{\text{Sigma}} \frac{\vec{J}(\vec{R}) \times (\vec{r}-\vec{R})}{c\,|\vec{r}-\vec{R}|^3}\,\mathrm{d}s$$

where $\vec{r}$ is the position vector of any point in space, $\vec{R}$ is the position vector of any point where the current exists, in this case a disk of radius a, and ds is the surface element. Sigma represents the integration domain and the above expression is a surface integral.

```
> restart;
  with(Physics[Vectors]);
  Setup(mathematicalnotation = true)
```

[&x, `+`, `.`, ChangeBasis, Component, Curl, DirectionalDiff, Divergence, Gradient, Identify, Laplacian, ∇, Norm, Setup, diff]

[mathematicalnotation = true] **(10.2.4.1.1)**

The expression for $\vec{H}$ can be entered as a double integral in cylindrical coordinates (rho, phi, z); the element of the surface of a disk in these coordinates is ρ dρ dφ, with rho varying from 0 to a and phi from 0 to 2 pi .

```
> H_ := Int(Int(J_&x (r_ - R_)/c/Norm(r_ - R_)^3 * rho, rho = 0..a), phi = 0..2
       * Pi);
```

$$\vec{H} := \int_0^{2\pi}\int_0^{a} \frac{\left(\vec{J}\times(\vec{r}-\vec{R})\right)\rho}{c\,\|\vec{r}-\vec{R}\|^3}\,\mathrm{d}\rho\,\mathrm{d}\phi \qquad \textbf{(10.2.4.1.2)}$$

We choose the system of references as in the previous problem, with the origin in the center of the disk, and the z axis oriented perpendicular to the disk. So again the position vector of a point over the z axis is

> $r\_ := z * \_k;$

$$\vec{r} := z\,\hat{k} \qquad \textbf{(10.2.4.1.3)}$$

and the position vector of a point of the disk is

> $R\_ := \mathrm{rho} * \_rho;$

$$\vec{R} := \rho\,\hat{\rho} \qquad \textbf{(10.2.4.1.4)}$$

By definition, the current $\vec{J}$ at a point $\vec{R}$ is equal to the value of the density of charge times the velocity of this charge; that is,

> $J\_ := \mathrm{sigma} * V\_;$

$$\vec{J} := \sigma\,\vec{V} \qquad \textbf{(10.2.4.1.5)}$$

Finally, the velocity $\vec{V}$ of a point $\vec{R}$ of the disk can be computed as the derivative of $\vec{R}$ with respect to t (the time), and in doing so we need to take into account that the unit vector $\widehat{\mathrm{rho}}$ *varies with time because it depends on the angle* φ and the disk is rotating.

This derivative of $\vec{R}$ can be computed in different ways. The simplest way is to state that the unit vector $\widehat{\mathrm{rho}}$ is a function of time t and differentiate:

> $\mathrm{rho} * \_rho(t);$

$$\rho\,\hat{\rho}(t) \qquad \textbf{(10.2.4.1.6)}$$

> $diff(\%, t);$

$$\rho\,\dot{\mathrm{phi}}(t)\,\hat{\phi}(t) \qquad \textbf{(10.2.4.1.7)}$$

Another way, step-by-step, is to make explicit this dependence of $\widehat{\mathrm{rho}}$ on phi by changing the basis onto which $\vec{R}$ is projected from the cylindrical to the Cartesian basis.

> $ChangeBasis(R\_, 1);$

$$\rho\cos(\phi)\,\hat{i} + \rho\sin(\phi)\,\hat{j} \qquad \textbf{(10.2.4.1.8)}$$

Now make phi depend on t and differentiate.

> $subs(\mathrm{phi} = \mathrm{phi}(t), \textbf{(10.2.4.1.8)});$

$$\rho\cos(\phi(t))\,\hat{i} + \rho\sin(\phi(t))\,\hat{j} \qquad \textbf{(10.2.4.1.9)}$$

> $diff(\textbf{(10.2.4.1.9)}, t)$

$$-\rho\,\dot{\mathrm{phi}}(t)\sin(\phi(t))\,\hat{i} + \rho\,\dot{\mathrm{phi}}(t)\cos(\phi(t))\,\hat{j} \qquad \textbf{(10.2.4.1.10)}$$

Introduce $\dot{\mathrm{phi}}(t) = \omega$, and remove the explicit dependence of phi with respect to t to arrive at an expression for $\vec{V}$.

> $factor(subs([diff(\mathrm{phi}(t), t) = \mathrm{omega}, \mathrm{phi}(t) = \mathrm{phi}], \textbf{??}))$

$$[\dot{\mathrm{phi}}(t) = \omega, \phi(t) = \phi] \qquad \textbf{(10.2.4.1.11)}$$

Yet another manner, knowing that $\widehat{\mathrm{rho}}$ and so $\vec{R} = \rho\ \widehat{\mathrm{rho}}$ depends on the time only through φ, you can compute $\vec{R} = \dot{\mathrm{phi}}\,\frac{d}{d\mathrm{phi}}\,\vec{R}(\mathrm{phi}) = \omega\,\frac{d}{d\mathrm{phi}}\,\vec{R}(\mathrm{phi})$. So

> $V\_ := \mathrm{omega} * diff(R\_, \mathrm{phi});$

$$\vec{V} := \omega\,\rho\,\hat{\phi} \qquad \textbf{(10.2.4.1.12)}$$

At this point, we have all the quantities defined in the system of coordinates chosen and in terms of the constant angular velocity, ω, and the radius of the disk, *a*. The expression of the magnetic field looks like

```
> H_;
```

$$\int_0^{2\pi}\int_0^{a} \frac{\left(\sigma\,\omega\,\rho^2\,\hat{k}+\sigma\,\omega\,\rho\,z\,\hat{\rho}\right)\rho}{c\left(\rho^2+z^2\right)^{3/2}}\,d\rho\,d\phi \qquad \textbf{(10.2.4.1.13)}$$

However, to perform the integrals, we still need to express $\widehat{\text{rho}}$ as a function of phi, one of the integration variables. For that purpose, it suffices to change the vectors involved ($\vec{R}$ and $\vec{V}$) to the Cartesian basis.

```
> R_ := ChangeBasis(R_, 1);
```

$$\vec{R} := \rho\cos(\phi)\,\hat{i}+\rho\sin(\phi)\,\hat{j} \qquad \textbf{(10.2.4.1.14)}$$

```
> V_ := ChangeBasis(V_, 1);
```

$$\vec{V} := -\omega\,\rho\sin(\phi)\,\hat{i}+\omega\,\rho\cos(\phi)\,\hat{j} \qquad \textbf{(10.2.4.1.15)}$$

With this change, $\vec{H}$ looks like

```
> H_;
```

$$\int_0^{2\pi}\int_0^{a} \frac{\left(\cos(\phi)\,\omega\,\rho\,\sigma\,z\,\hat{i}+\sin(\phi)\,\omega\,\rho\,\sigma\,z\,\hat{j}+\sigma\,\omega\,\rho^2\,\hat{k}\right)\rho}{c\left(\rho^2\cos(\phi)^2+\rho^2\sin(\phi)^2+z^2\right)^{3/2}}\,d\rho\,d\phi \qquad \textbf{(10.2.4.1.16)}$$

and so the integrals can be performed, leading to the desired value of the magnetic field $\vec{H}$ on the axis of the rotating disk.

```
> H_ := value(H_) assuming a > 0, z > 0;
```

$$\vec{H} := -\frac{2\,\omega\,\sigma\,\pi\,\hat{k}\left(2\sqrt{a^2+z^2}\,z-a^2-2\,z^2\right)}{\sqrt{a^2+z^2}\,c} \qquad \textbf{(10.2.4.1.17)}$$

```
>
```

7. Tensors and Special Relativity

Algebraic representation	Indexed objects, as A[x], can include functionality as in A[x](x), must be defined using the Physics:-Define command and can have (anti)symmetry properties of any particular kind regarding permutations of indices. You can also define tensors using tensorial equations.
Commands	Physics:-Define, Physics:-Simplify, and Physics:-`.` to "multiply and simplify" in one step - useful to directly perform a contraction instead of just representing it.

Table 7: Tensors

Examples

```
> restart;
  with(Physics) :
  Setup(mathematicalnotation = true);
```

$$[mathematicalnotation = true] \quad \textbf{(11.1.1)}$$

To define a system of references (coordinates system) and the related spacetime vector you can use Setup or Coordinates;

```
> Coordinates(X);
```

Default differentiation variables for d_, D_ and dAlembertian are: $\{X = (x1, x2, x3, x4)\}$

Systems of spacetime Coordinates are: $\{X = (x1, x2, x3, x4)\}$

$$\{X\} \quad \textbf{(11.1.2)}$$

You can now use X to represent function dependency, as in $F(X)$, equivalent to writing $F(x1, x2, x3, x4)$, and also as the spacetime vector X^{mu} (see also SpaceTimeVector). To indicate that an index is contravariant, prefix it with ~ (tilde)

```
> X[~mu], X[mu];
```

$$X^{\mu}, X_{\mu} \quad \textbf{(11.1.3)}$$

The label X can also be used to select each of the components of the spacetime vector

```
> X[1], X[3];
```

$$x1, x3 \quad \textbf{(11.1.4)}$$

and $x0$ is always mapped into $x4$

```
> x0;
```

$$x4 \quad \textbf{(11.1.5)}$$

You can also set the coordinates to be any sequence of four names. Three predefined sets are 'cartesian', 'cylindrical' and 'spherical'

```
> Coordinates(X = cartesian)
```

Default differentiation variables for d_, D_ and dAlembertian are: $\{X = (x, y, z, t)\}$

Systems of spacetime Coordinates are: $\{X = (x, y, z, t)\}$

$$\{X\} \quad \textbf{(11.1.6)}$$

In all cases you will still be able to refer to each coordinate using the generic symbols $x1$, $x2$, ...

```
> x1, x2;
```

$$x, y \quad \textbf{(11.1.7)}$$

```
> x0 = x4;
```

$$t = t \quad \textbf{(11.1.8)}$$

You can set many systems of coordinates simultaneously - although only one is considered the 'default differentiation variables' for the d_ dAlembertian and D_ operators (and that is the system of references were are all the general relativity tensors are defined).

```
> Coordinates(Y);
```

Systems of spacetime Coordinates are: $\{X = (x, y, z, t), Y = (y1, y2, y3, y4)\}$

$$\{X, Y\} \quad \textbf{(11.1.9)}$$

```
> y0 = y4;
```

$$y4 = y4 \qquad \textbf{(11.1.10)}$$

```
> Setup(diff);
```

** Partial match of 'Physics:-diff' against keyword 'differentiationvariables'*

Default differentiation variables for d_, D_ and dAlembertian are: $\{X = (x, y, z, t)\}$

$$[differentiationvariables = [X]] \qquad \textbf{(11.1.11)}$$

To change the differentiation variables enter for instance **Setup(diff = Y);**

When you load Physics, some package's commands that are automatically set as spacetime tensors, such as the metric g_[mu, nu] or the differential operator d_[mu]. Every other symbol that you want to be considered a spacetime tensor must be defined as such using the Define command, and **nothing else** will be considered a spacetime tensor during computations. To see the tensors predefined you can call Define without arguments

```
> Define( );
```

Defined as tensors

$$\{\gamma_{\mu}, \sigma_{\mu}, X_{\mu}, Y_{\mu}, \partial_{\mu}, g_{\mu,\nu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \qquad \textbf{(11.1.12)}$$

This set includes the special relativity tensors as well as the Pauli and Dirac matrices. In curved spacetimes the returned set includes also the general relativity tensors.

The default dimension of spacetime is

```
> Setup(dimension);
```

$$[dimension = 4] \qquad \textbf{(11.1.13)}$$

To change the value of the spacetime dimension use **Setup(dimension = N)** where N is any positive integer greater than 1.

To see the values of the components of the metric you can enter the spacetime metric g_ without indices; when you load the package it is of Minkowski type

```
> g_[ ];
```

$$g_{\mu,\nu} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(11.1.14)}$$

The spacetime indexed differentiation operator

```
> d_[mu](F(X));
```

$$\partial_{\mu}(F(X)) \qquad \textbf{(11.1.15)}$$

This operator is also a representation for the *differential* of a function; for that purpose enter it without indices

```
> d_(F(X));
```

$$(\partial_{\mu}(F(X)))\,(\partial(X^{\mu})) \qquad \textbf{(11.1.16)}$$

Using d_ you can express the differential of any coordinate defined using Coordinates

```
> d_(x0);
```

$$\partial(t) \qquad \textbf{(11.1.17)}$$

```
> d_(y0);
```

$$\partial(y4) \qquad \textbf{(11.1.18)}$$

The differential of everything else is zero

```
> d_(a);
```

$$0 \qquad \textbf{(11.1.19)}$$

The diff and d_ operators are interconnected

```
> diff(F(X), X[~mu]);
```

$$\partial_{\mu}(F(X)) \qquad \textbf{(11.1.20)}$$

```
> diff(F(X), X[mu]);
```

$$\partial^{\mu}(F(X)) \qquad \textbf{(11.1.21)}$$

By applying it two times you get the dAlembertian

```
> d_[mu](d_[mu](F(X)));
```

$$\Box(F(X)) \qquad \textbf{(11.1.22)}$$

You can enter any pair of contracted indices one covariant and one contravariant, not being relevant which one is of which kind, so you can also enter both covariant or both contravariant and the system will automatically rewrite them as one covariant and one contravariant.

```
> g_[mu, nu]^2;
```

$$g_{\mu,\nu}\, g^{\mu,\nu} \qquad \textbf{(11.1.23)}$$

Define two tensors A and B for exploration purposes

```
> Define(A, B);
```

Defined objects with tensor properties

$$\{A, B, \gamma_{\mu}, \sigma_{\mu}, X_{\mu}, Y_{\mu}, \partial_{\mu}, g_{\mu,\nu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \qquad \textbf{(11.1.24)}$$

Note the automatic rewriting of repeated indices as one covariant and one contravariant

```
> g_[alpha, mu] * A[mu] * g_[alpha, nu] * B[nu, sigma, sigma];
```

$$B_{\nu,\sigma}{}^{\sigma}\, g_{\alpha,\mu}\, g^{\alpha,\nu}\, A^{\mu} \qquad \textbf{(11.1.25)}$$

When a pair of contracted indices is already one covariant and one contravariant, it is left unchanged. Although it is not recommended, you can switch off the Physics evaluator that performs this automatic rewriting of contracted indices by entering **Setup(usephysicsevaluator = false)**.

To determine the repeated and free indices in an expression use Check

```
> Check((11.1.25), all)
```

The repeated indices per term are: $[\{\ldots\}, \{\ldots\}, \ldots]$*; the free indices are:* $\{\ldots\}$

$$[\{\alpha, \mu, \nu, \sigma\}], \{\} \qquad \textbf{(11.1.26)}$$

The main simplifier in the Physics package is Simplify. The simplification of tensor indices takes into account the sum rule for repeated indices

```
> Simplify((11.1.25))
```

(11.1.27)

$$B_{\nu,\sigma}{}^{\sigma} A^{\nu} \qquad \textbf{(11.1.27)}$$

Besides its use in quantum mechanics, the Physics `.` operator is also a handy shortcut for simplifying contracted indices - you use it replacing the product operator *. This is useful when you want the simplification to happen only in some places. Recalling that multiplication is left associative, you may need to put parenthesis to get what you want. For example, replace in **??** only the third `*` by `.`

```
> g_[alpha, mu] * A[mu] * (g_[alpha, nu] . B[nu, sigma, sigma]);
```

$$B_{\alpha,\sigma}{}^{\sigma} g^{\alpha}{}_{\mu} A^{\mu} \qquad \textbf{(11.1.28)}$$

Exercises

1. Define a contravariant 4-vector p^{mu} having for components $[p_x, p_y, p_z, p_t]$

Solution

```
> restart; with(Physics):
> p[~mu] = [p_x, p_y, p_z, p_t]
```

$$p_{\sim mu} = [p_x, p_y, p_z, p_t] \qquad \textbf{(11.2.1.1)}$$

```
> Define((11.2.1.1))
```

Defined objects with tensor properties

$$\{\gamma_{\mu}, \sigma_{\mu}, \partial_{\mu}, g_{\mu,\nu}, p^{\mu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \qquad \textbf{(11.2.1.2)}$$

You can now use p[mu] to represent this tensor. For example, compute the value of p^2

```
> p[mu]^2
```

$$p_{\mu} p^{\mu} \qquad \textbf{(11.2.1.3)}$$

```
> SumOverRepeatedIndices((11.2.1.3))
```

$$p_t{}^2 - p_x{}^2 - p_y{}^2 - p_z{}^2 \qquad \textbf{(11.2.1.4)}$$

You can also retrieve its components giving values to the index, these are the covariant and contravariant first components

```
> p[1]
```

$$-p_x \qquad \textbf{(11.2.1.5)}$$

```
> 'p[~1]'
```

$$p^1 \qquad \textbf{(11.2.1.6)}$$

```
> %
```

$$p_x \qquad \textbf{(11.2.1.7)}$$

```
>
```

2. Define two tensors A[mu] and B[alpha, beta] and
   a) Symmetrize their product
   b) Define B as symmetric and symmetrize again

c) Define C[alpha,beta,gamma] as totally antisymmetric, antisymmetrize the product of A with C and show that due to the antisymmetry of C it contains only 4 terms

Solution

```
> restart; with(Physics):
> Define(A, B)
```

Defined objects with tensor properties

$$\{A, B, \gamma_{\mu}, \sigma_{\mu}, \partial_{\mu}, g_{\mu,\nu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \qquad \textbf{(11.2.2.1)}$$

```
> A[mu] B[alpha, beta]
```

$$A_{\mu} B_{\alpha,\beta} \qquad \textbf{(11.2.2.2)}$$

```
> Symmetrize((11.2.2.2))
```

$$\frac{1}{6} A_{\alpha} B_{\beta,\mu} + \frac{1}{6} A_{\alpha} B_{\mu,\beta} + \frac{1}{6} A_{\beta} B_{\alpha,\mu} + \frac{1}{6} A_{\beta} B_{\mu,\alpha} + \frac{1}{6} A_{\mu} B_{\alpha,\beta} + \frac{1}{6} A_{\mu} B_{\beta,\alpha} \qquad \textbf{(11.2.2.3)}$$

This result cannot be simplified further because there are no repeated indices and no symmetries for the indices

```
> Simplify((11.2.2.3))
```

$$\frac{1}{6} A_{\alpha} B_{\beta,\mu} + \frac{1}{6} A_{\alpha} B_{\mu,\beta} + \frac{1}{6} A_{\beta} B_{\alpha,\mu} + \frac{1}{6} A_{\beta} B_{\mu,\alpha} + \frac{1}{6} A_{\mu} B_{\alpha,\beta} + \frac{1}{6} A_{\mu} B_{\beta,\alpha} \qquad \textbf{(11.2.2.4)}$$

Defining B as symmetric, this result can be simplified

```
> Define(B, symmetric)
```

Defined objects with tensor properties

$$\{A_{\mu}, B_{\alpha,\beta}, \gamma_{\mu}, \sigma_{\mu}, \partial_{\mu}, g_{\mu,\nu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \qquad \textbf{(11.2.2.5)}$$

```
> Simplify((11.2.2.3))
```

$$\frac{A_{\alpha} B_{\beta,\mu}}{3} + \frac{A_{\beta} B_{\alpha,\mu}}{3} + \frac{A_{\mu} B_{\alpha,\beta}}{3} \qquad \textbf{(11.2.2.6)}$$

Or directly

```
> Symmetrize((11.2.2.2), simplifytensor)
```

$$\frac{A_{\alpha} B_{\beta,\mu}}{3} + \frac{A_{\beta} B_{\alpha,\mu}}{3} + \frac{A_{\mu} B_{\alpha,\beta}}{3} \qquad \textbf{(11.2.2.7)}$$

```
> Define(C[mu, nu, rho], antisymmetric);
```

Defined objects with tensor properties

$$\{A_{\mu}, B_{\alpha,\beta}, C_{\mu,\nu,\rho}, \gamma_{\mu}, \sigma_{\mu}, \partial_{\mu}, g_{\mu,\nu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \qquad \textbf{(11.2.2.8)}$$

```
> A[alpha] C[mu, nu, rho]
```

$$A_{\alpha} C_{\mu,\nu,\rho} \qquad \textbf{(11.2.2.9)}$$

> *Antisymmetrize*(**(11.2.2.9)**)

$$\frac{1}{24} A_\alpha C_{\mu,\nu,\rho} - \frac{1}{24} A_\alpha C_{\mu,\rho,\nu} - \frac{1}{24} A_\alpha C_{\nu,\mu,\rho} + \frac{1}{24} A_\alpha C_{\nu,\rho,\mu} + \frac{1}{24} A_\alpha C_{\rho,\mu,\nu} - \frac{1}{24} A_\alpha C_{\rho,\nu,\mu} - \frac{1}{24} A_\mu C_{\alpha,\nu,\rho} + \frac{1}{24} A_\mu C_{\alpha,\rho,\nu} + \frac{1}{24} A_\mu C_{\nu,\alpha,\rho} - \frac{1}{24} A_\mu C_{\nu,\rho,\alpha} - \frac{1}{24} A_\mu C_{\rho,\alpha,\nu} + \frac{1}{24} A_\mu C_{\rho,\nu,\alpha} + \frac{1}{24} A_\nu C_{\alpha,\mu,\rho} - \frac{1}{24} A_\nu C_{\alpha,\rho,\mu} - \frac{1}{24} A_\nu C_{\mu,\alpha,\rho} + \frac{1}{24} A_\nu C_{\mu,\rho,\alpha} + \frac{1}{24} A_\nu C_{\rho,\alpha,\mu} - \frac{1}{24} A_\nu C_{\rho,\mu,\alpha} - \frac{1}{24} A_\rho C_{\alpha,\mu,\nu} + \frac{1}{24} A_\rho C_{\alpha,\nu,\mu} + \frac{1}{24} A_\rho C_{\mu,\alpha,\nu} - \frac{1}{24} A_\rho C_{\mu,\nu,\alpha} - \frac{1}{24} A_\rho C_{\nu,\alpha,\mu} + \frac{1}{24} A_\rho C_{\nu,\mu,\alpha} \quad \textbf{(11.2.2.10)}$$

> *Simplify*(**(11.2.2.10)**)

$$\frac{A_\alpha C_{\mu,\nu,\rho}}{4} - \frac{A_\mu C_{\alpha,\nu,\rho}}{4} + \frac{A_\nu C_{\alpha,\mu,\rho}}{4} - \frac{A_\rho C_{\alpha,\mu,\nu}}{4} \quad \textbf{(11.2.2.11)}$$

>

3. Compute Maxwell equations departing from the Action for electrodynamics

Solution

Maxwell equations result from taking the functional derivative of the Action.

Let X and Y represent two spacetime points

> *restart*; *with*(*Physics*) :

> *Coordinates*(*X*, *Y*)

Default differentiation variables for d_, D_ and dAlembertian are: {*X* = (*x1*, *x2*, *x3*, *x4*)}

Systems of spacetime Coordinates are: {*X* = (*x1*, *x2*, *x3*, *x4*), *Y* = (*y1*, *y2*, *y3*, *y4*)}

$$\{X, Y\} \quad \textbf{(11.2.3.1)}$$

Define now the 4-D electromagnetic potential

> *Define*(*A*[mu](*X*));

Defined objects with tensor properties

$$\{A_\mu, \gamma_\mu, \sigma_\mu, X_\mu, Y_\mu, \partial_\mu, g_{\mu,\nu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \quad \textbf{(11.2.3.2)}$$

From herein avoid *displaying* the functionality whenever it is X

> *PDEtools*:-*declare*(*A*(*X*));

A(*x1*, *x2*, *x3*, *x4*) *will now be displayed as A* **(11.2.3.3)**

Also to avoid having to repeat saying who are the differentiation variables of the operators ∂_{mu} and $\Box$ just set them

```
> Setup(var = X);
```

```
        * Partial match of  'var' against keyword 'differentiationvariables'
Default differentiation variables for d_, D_ and dAlembertian are: {X = (x1, x2,
    x3, x4)}
```

$$[differentiationvariables = [X]] \quad \textbf{(11.2.3.4)}$$

Query about the dimension and signature ...

```
> Setup(dimension, signature);
```

$$[dimension = 4, signature = `-`] \quad \textbf{(11.2.3.5)}$$

Everything is set. Let's define the electromagnetic field tensor $F_{mu,\,nu}$ in terms of the derivatives of A_{mu}

```
> F[mu, nu] := d_[mu](A[nu](X)) - d_[nu](A[mu](X));
```

$$F_{\mu,\,\nu} := \partial_{\mu}\left(A_{\nu}\right) - \left(\partial_{\nu}\left(A_{\mu}\right)\right) \quad \textbf{(11.2.3.6)}$$

Define now the Action

```
> S := Intc(F[mu, nu]^2, X);
```

$$S := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\partial_{\mu}\left(A_{\nu}\right) - \left(\partial_{\nu}\left(A_{\mu}\right)\right)\right)^{2}\,dx1\,dx2\,dx3\,dx4 \quad \textbf{(11.2.3.7)}$$

Take now the functional derivative of S (use ' to delay the computation, to see what is that will be computed)

```
> 'Fundiff'(S, A[rho]);
```

$$\left(\frac{\delta}{\delta A_{\rho}}\right)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\partial_{\mu}\left(A_{\nu}\right) - \left(\partial_{\nu}\left(A_{\mu}\right)\right)\right)^{2}\,dx1\,dx2\,dx3\,dx4 \quad \textbf{(11.2.3.8)}$$

```
> (11.2.3.8)
```

$$\left(2\left(\partial_{\mu}\left(\partial_{\nu}\left(A^{\nu}\right)\right)\right) - 2\,\Box\left(A_{\mu}\right)\right)g^{\mu,\,\rho} + \left(-2\,\Box\left(A_{\nu}\right) + 2\left(\partial_{\mu}\left(\partial_{\nu}\left(A^{\mu}\right)\right)\right)\right)g^{\nu,\,\rho} \quad \textbf{(11.2.3.9)}$$

To simplify the contracted spacetime indices, use the Simplify command, resulting in the Maxwell equations in their familiar 4-D tensorial form

```
> Simplify(%)
```

$$-4\,\Box\left(A^{\rho}\right) + 4\left(\partial_{\mu}\left(\partial^{\rho}\left(A^{\mu}\right)\right)\right) \quad \textbf{(11.2.3.10)}$$

4. Compute the geodesics (the shortest line joining two points in space) for a Minkowski spacetime

Solution

```
> restart; with(Physics):
> Setup(coordinates = X);
```

```
        * Partial match of  'coordinates' against keyword 'coordinatesystems'
Default differentiation variables for d_, D_ and dAlembertian are: {X = (x1, x2,
    x3, x4)}
        Systems of spacetime Coordinates are: {X = (x1, x2, x3, x4)}
```

(11.2.4.1)

$$[coordinatesystems = \{X\}] \tag{11.2.4.1}$$

The spacetime metric at this point is the one that loads with Physics, i.e. a Minkowski type metric

> $g\_[\,]$

$$g_{\mu,\nu} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11.2.4.2}$$

The geodesics equations for this metric, first as an algebraic equation in tensor notation, then as a list of equations for the tensorial components

> $Geodesics(tensornotation)$;

$$\frac{d^2}{d\tau^2} X^{\mu}(\tau) \tag{11.2.4.3}$$

> $Geodesics()$;

$$\left[\frac{d^2}{d\tau^2} x4(\tau) = 0, \frac{d^2}{d\tau^2} x3(\tau) = 0, \frac{d^2}{d\tau^2} x2(\tau) = 0, \frac{d^2}{d\tau^2} x1(\tau) = 0\right] \tag{11.2.4.4}$$

The straight-lines solution computed directly

> $Geodesics(output = solutions)$;

$$\{x1(\tau) = \_C1\,\tau + \_C2, x2(\tau) = \_C3\,\tau + \_C4, x3(\tau) = \_C5\,\tau + \_C6, x4(\tau) = \_C7\,\tau + \_C8\} \tag{11.2.4.5}$$

5. The Killing vectors generate transformations that leave the metric g_ invariant in form. Show that the components of the Killing vector for a Minkowski spacetime generates translations along the four axes.

Solution

> $restart; with(Physics)$:

The default spacetime is of Minkowski kind

> $g\_[\,]$

$$g_{\mu,\nu} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11.2.5.1}$$

Set a system of Coordinates

> $Setup(coordinates = X)$;

** Partial match of 'coordinates' against keyword 'coordinatesystems'*

Default differentiation variables for d_, D_ and dAlembertian are: $\{X = (x1, x2, x3, x4)\}$

$$\textit{Systems of spacetime Coordinates are: } \{X = (x1, x2, x3, x4)\}$$

$$[\textit{coordinatesystems} = \{X\}] \qquad \textbf{(11.2.5.2)}$$

Define a tensor with one index to represent the Killing vector

```
> Define(V);
```

$$\textit{Defined objects with tensor properties}$$

$$\{V, \gamma_{\mu}, \sigma_{\mu}, X_{\mu}, \partial_{\mu}, g_{\mu,\nu}, \delta_{\mu,\nu}, \varepsilon_{\alpha,\beta,\mu,\nu}\} \qquad \textbf{(11.2.5.3)}$$

The covariant components V_{alpha}

```
> KillingVectors(V, output = equations);
```

$$\left[\frac{\partial}{\partial x1} V_4(X) = -\left(\frac{\partial}{\partial x4} V_1(X)\right), \frac{\partial}{\partial x2} V_4(X) = -\left(\frac{\partial}{\partial x4} V_2(X)\right), \frac{\partial}{\partial x3} V_4(X) \right. \qquad \textbf{(11.2.5.4)}$$

$$= -\left(\frac{\partial}{\partial x4} V_3(X)\right), \frac{\partial}{\partial x4} V_4(X) = 0, \frac{\partial^2}{\partial x4^2} V_3(X) = 0, \frac{\partial}{\partial x1} V_3(X) =$$

$$-\left(\frac{\partial}{\partial x3} V_1(X)\right), \frac{\partial}{\partial x2} V_3(X) = -\left(\frac{\partial}{\partial x3} V_2(X)\right), \frac{\partial}{\partial x3} V_3(X) = 0, \frac{\partial^2}{\partial x3^2}$$

$$V_2(X) = 0, \frac{\partial^2}{\partial x3 \partial x4} V_2(X) = 0, \frac{\partial^2}{\partial x4^2} V_2(X) = 0, \frac{\partial}{\partial x1} V_2(X) = -\left(\frac{\partial}{\partial x2}\right.$$

$$V_1(X)\Big), \frac{\partial}{\partial x2} V_2(X) = 0, \frac{\partial^2}{\partial x2^2} V_1(X) = 0, \frac{\partial^2}{\partial x2 \partial x3} V_1(X) = 0, \frac{\partial^2}{\partial x2 \partial x4}$$

$$V_1(X) = 0, \frac{\partial^2}{\partial x3^2} V_1(X) = 0, \frac{\partial^2}{\partial x3 \partial x4} V_1(X) = 0, \frac{\partial^2}{\partial x4^2} V_1(X) = 0, \frac{\partial}{\partial x1}$$

$$\left. V_1(X) = 0\right]$$

```
> KillingVectors(V);
```

$$\{V_1(X) = \_C1\, x2 + \_C2\, x3 + \_C3\, x4 + \_C4, V_2(X) = -\_C1\, x1 + \_C5\, x3 \qquad \textbf{(11.2.5.5)}$$

$$+ \_C6\, x4 + \_C7, V_3(X) = -\_C2\, x1 - \_C5\, x2 + \_C8\, x4 + \_C9, V_4(X) =$$

$$-\_C3\, x1 - \_C6\, x2 - \_C8\, x3 + \_C10\}$$

To understand this result, recall that the Killing vectors generate transformations that leave the metric g_ invariant in form; these are isometries. The result **(11.2.5.5)** generates translations along the four axis that leave invariant the distance between two points and thus invariant in form the metric.

```
>
```

8. Quantum Mechanics

Commands	*Annihilation*, *AntiCommutator*, *Bra*, *Bracket*, *Check*, *Coefficients*, *Commutator*, *Creation*, *Dagger*, *Dgamma*, *Fundiff*, *GrassmannParity*, *Gtaylor*, *Intc*, *Inverse*,

	Ket, *KroneckerDelta*, *LeviCivita*, *Normal*, *Parameters*, *PerformOnAnticommutativeSystem*, *Projector*, *Psigma*, *Setup*, *Simplify*, *ToFieldComponents*, *ToSuperfields*, *Trace*, *dAlembertian*, *d_* To apply an operator to a Ket, or to take the scalar product of a Bra with a Ket use the dot operator .
Manipulation commands	Setup and Simplify

Table 8: Quantum Mechanics

Dirac Notation

Kets and Bras

```
> restart; with(Physics):
```

The quantum state of a system, belonging to a space of quantum states, is represented in Dirac notation by a Ket state-vector.

```
> Ket(u);
```

$$|u\rangle \qquad \textbf{(12.1.1.1)}$$

The above is the quantum analog of a non-projected vector $\vec{u}$ of a 3-D Euclidean space.

- The norm of a generic Ket $|u\rangle$ is not predefined and can be indicated by setting a bracket rule, as shown below.
- Every **Ket** can be projected onto a basis of state-vectors
- The **Kets** conforming a basis (analogous to 3-D unit vectors) are distinguished from a generic Ket $|u\rangle$ by the fact that they have one or many *quantum numbers,* as in

```
> Ket(v, n);
```

$$|v_n\rangle \qquad \textbf{(12.1.1.2)}$$

Kets having quantum numbers are always assumed to belong to a basis of quantum states, are orthogonal to each other, have norm equal to 1, and are distinguished from each other by the values of these quantum numbers. For example, an orthonormal basis of a two dimensional space of quantum states is

```
> Ket(v, 0), Ket(v, 1);
```

$$|v_0\rangle, |v_1\rangle \qquad \textbf{(12.1.1.3)}$$

There are no restrictions on the number of *quantum numbers* that a **Ket** can have. This is a **Ket** belonging to a basis of a space that depends on four quantum numbers.

```
> Ket(v, j, k, m, n);
```

$$|v_{j,\,k,\,m,\,n}\rangle \qquad \textbf{(12.1.1.4)}$$

You can associate a space of states with each quantum number, so a **Ket** with many quantum numbers represents a state in a space constructed as a *tensor product of spaces*.

There is a Bra associated with each **Ket**, obtained from the **Ket** by performing the Hermitian

conjugate, or Dagger, operation.

```
> %Dagger(%);
```

$$|v_{j,k,m,n}\rangle^{\dagger} \quad \textbf{(12.1.1.5)}$$

```
> value(%)
```

$$\langle v_{j,k,m,n}| \quad \textbf{(12.1.1.6)}$$

```
> Dagger(%);
```

$$|v_{j,k,m,n}\rangle \quad \textbf{(12.1.1.7)}$$

You can enter **Bras** directly by using the **Bra** function.

```
> Bra(v,j,k,m,n);
```

$$\langle v_{j,k,m,n}| \quad \textbf{(12.1.1.8)}$$

The space of **Bras** of a system is the dual of the space of **Kets** of that system.

Discrete and continuous basis of states

Kets belong to either a discrete or a continuous spaces of quantum states.

A discrete space of states is one where the quantum numbers of its **Kets** vary discretely. These **Kets** thus belong to *discrete bases of quantum states*.

A continuous space is one where the quantum numbers vary continuously and its **Kets** belong to *continuous bases of quantum states*.

Unless explicitly stated otherwise, **Kets** are assumed to belong to discrete space of states.

You can indicate that a label *R* identifies a continuous space of states by using the Setup command.

```
> Setup(continuous = R);
```

```
   * Partial match of 'continuous' against keyword 'quantumcontinuousbasis'
```

$$[quantumcontinuousbasis = \{R\}] \quad \textbf{(12.1.2.1)}$$

Kets of a continuous space of states can also have any number of *quantum numbers*.

```
> Ket(R,x);
```

$$|R_x\rangle \quad \textbf{(12.1.2.2)}$$

```
> Ket(R,x,y,z);
```

$$|R_{x,y,z}\rangle \quad \textbf{(12.1.2.3)}$$

```
> Dagger(%);
```

$$\langle R_{x,y,z}| \quad \textbf{(12.1.2.4)}$$

Scalar product and orthonormalization relation

The scalar product is defined between a Bra and a Ket, in that order, and can be performed by using the dot operator `.` of the Physics package, or by using the Bracket function; both represent the same object.

```
> Bra(u) . Ket(v);
```

$$\langle u \mid v \rangle \qquad \textbf{(12.1.3.1)}$$

```
> Bracket(Bra(u), Ket(v));
```

$$\langle u \mid v \rangle \qquad \textbf{(12.1.3.2)}$$

Note that when the scalar product is just represented, not actually computed, as in the above, the result is always expressed in terms of the **Bracket** function. A shortcut notation for entering the scalar product using the **Bracket** function is

```
> Bracket(u, v);
```

$$\langle u \mid v \rangle \qquad \textbf{(12.1.3.3)}$$

Under the Dagger operation, $(A \cdot B)^{\dagger} = B^{\dagger} \cdot A^{\dagger}$ so that the **Bracket** becomes

```
> Dagger(%);
```

$$\langle v \mid u \rangle \qquad \textbf{(12.1.3.4)}$$

For the **Bracket**, the same happens under conjugation:

```
> conjugate(%);
```

$$\langle u \mid v \rangle \qquad \textbf{(12.1.3.5)}$$

Two generic **Kets** such as the above may or not belong to the same space. To make practical use of **Kets**, depending on the problem you may want to set a *bracket rule*, stating the value of the **Bracket** between them, by using the Setup command.

```
> Setup(% = f(u, v));
```

$$[bracketrules = \{\langle u \mid v \rangle = f(u, v)\}] \qquad \textbf{(12.1.3.6)}$$

After that, both the *dot operator* of the **Physics** package and the **Bracket** function know how to perform their scalar product.

```
> Bracket(u, v);
```

$$f(u, v) \qquad \textbf{(12.1.3.7)}$$

Kets belonging to a discrete basis of states satisfy an ***orthonormalization relation*** involving the KroneckerDelta symbol, and when the quantum numbers are present, you do not need to specify a bracket rule.

```
> %Bracket(Bra(u, n) , Ket(u, m)) = Bra(u, n) . Ket(u, m);
```

$$\langle u_n \mid u_m \rangle = \delta_{m, n} \qquad \textbf{(12.1.3.8)}$$

Note in the above the inert form **%Bracket**. It is sometimes useful to represent mathematical operations without having them actually performed. For that purpose, use any command with the name prefixed by the symbol **%**. To have the inert operation performed, use the value command.

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

$$\delta_{m, n} = \delta_{m, n} \qquad \textbf{(12.1.3.9)}$$

Evaluate the output two operations above at $m = n$.

```
> eval(%%, m = n);
```

$$\langle u_n \mid u_n \rangle = 1 \qquad \textbf{(12.1.3.10)}$$

The **Bracket** of state-vectors depending on many quantum numbers results in products of **KroneckerDelta** symbols. The shortcut notation of the **Bracket** function also works in the presence of quantum numbers (**tip**: to avoid typographical mistakes, it is practical to group the objects visually, by leaving spaces or not after the commas).

> *Bracket*(*u*, *i*, *j*, *k*, *u*, *n*, *m*, *l*);

$$\delta_{i,n}\,\delta_{j,m}\,\delta_{k,l} \qquad \textbf{(12.1.3.11)}$$

Kets of a continuous basis of states satisfy an ***orthonormalization relation*** involving the Dirac function (recall that *R* has been set as a label of a continuous space, by using the Setup command, above).

> *Bracket*(*R*, *x*, *R*, *y*);

$$\delta(y-x) \qquad \textbf{(12.1.3.12)}$$

> *%Bracket*(*R*, *x*, *y*, *z*, *R*, *a*, *b*, *c*) :
> *% = value*(*%*);

$$\langle R_{x,y,z} \mid R_{a,b,c} \rangle = \delta^{(3)}([a-x, b-y, c-z]) \qquad \textbf{(12.1.3.13)}$$

In the above, the 3-dimensional Dirac function can be expanded using expand

> *expand*(*%*)

$$\langle R_{x,y,z} \mid R_{a,b,c} \rangle = \delta(a-x)\,\delta(b-y)\,\delta(c-z) \qquad \textbf{(12.1.3.14)}$$

Closure relation, Projectors

Every Ket of a space of states can be expanded into a basis of that space. The operator that performs the expansion is called a Projector. To construct these projectors, information about the basis dimension is necessary. You can indicate this dimension directly to the **Projector** command, or set it by using Setup. The information available at this point is

> *Setup*(*basisdim*, *cont*);

** Partial match of 'basisdim' against keyword 'quantumbasisdimension'*

** Partial match of 'cont' against keyword 'quantumcontinuousbasis'*

[quantumbasisdimension = none, quantumcontinuousbasis = {R}] **(12.1.4.1)**

By default, continuous bases are assumed to range from $-\infty$ to ∞, so this information on *R* is enough to compute its **Projector**.

> *P*[*R*] := *Projector*(*Ket*(*R*, *x*, *y*, *z*));

$$P_R := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |R_{x,y,z}\rangle \langle R_{x,y,z}|\,dx\,dy\,dz \qquad \textbf{(12.1.4.2)}$$

This expression P_R for the projector is also called the ***closure relation***; together with the ***orthonormalization relation*** $\langle R_{x,y,z} \mid R_{a,b,c} \rangle = \text{delta}^{(3)}([a-x, b-y, c-z])$, it tells that the set of Kets $\{|R_{x,y,z}\rangle\}$ containing all the possible values of *x*, *y*, and *z* forms a basis, and so any | psi ⟩ has a unique expansion onto $\{|R_{x,y,z}\rangle\}$.

Note that the scalar product of P_R with itself is equal to itself.

> *P*[*R*] . *P*[*R*];

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |R_{x,y,z}\rangle \langle R_{x,y,z}|\,dx\,dy\,dz \qquad \textbf{(12.1.4.3)}$$

The following is the projector for a basis generically labeled *u* that has not been set to represent a continuous basis. By default, if nothing is known about the label of a basis, it is assumed to

be related to a discrete space of states. The dimension of the basis can be indicated directly to the **Projector** command.

```
> P[u] := Projector(Ket(u, n), dimension = N);
```

$$P_u := \sum_{n=0}^{N-1} |u_n\rangle \langle u_n| \qquad \textbf{(12.1.4.4)}$$

The information passed to **Projector** is automatically tracked by the system, so you do not need to give it again.

```
> Setup(quantumbasisdimension, quantumcontinuousbasis);
```

$$[quantumbasisdimension = \{R = \infty, u = N\}, quantumcontinuousbasis = \{R\}] \qquad \textbf{(12.1.4.5)}$$

To change this information, see Setup and its *redo* option.

In order to compute scalar products of Kets belonging to a basis with other **Kets** of the same space, you can define a bracketrule.

```
> %Bracket(Bra(R, x, y, z), Ket(psi)) = psi(x, y, z);
```

$$\langle R_{x,y,z} | \psi \rangle = \psi(x, y, z) \qquad \textbf{(12.1.4.6)}$$

```
> Setup(%);
```

$$[bracketrules = \{\langle u | v \rangle = f(u, v), \langle R_{x,y,z} | \psi \rangle = \psi(x, y, z)\}] \qquad \textbf{(12.1.4.7)}$$

Now Bracket and the `.` operator of the Physics package know how to compute a number of related operations.

```
> Bracket(Bra(R, a, b, c), Ket(psi));
```

$$\psi(a, b, c) \qquad \textbf{(12.1.4.8)}$$

```
> P[R] . Ket(psi);
```

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \psi(x, y, z) |R_{x,y,z}\rangle \, dx\, dy\, dz \qquad \textbf{(12.1.4.9)}$$

```
> Bra(psi) . P[R];
```

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \overline{\psi(x, y, z)} \langle R_{x,y,z}| \, dx\, dy\, dz \qquad \textbf{(12.1.4.10)}$$

```
> Bra(psi) . P[R] . Ket(psi);
```

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |\psi(x, y, z)|^2 \, dx\, dy\, dz \qquad \textbf{(12.1.4.11)}$$

```
> %Bracket(Bra(R, x, y, z), Ket(phi)) = phi(x, y, z)
```

$$\langle R_{x,y,z} | \phi \rangle = \phi(x, y, z) \qquad \textbf{(12.1.4.12)}$$

```
> Setup(%)
```

$$[bracketrules = \{\langle u | v \rangle = f(u, v), \langle R_{x,y,z} | \phi \rangle = \phi(x, y, z), \langle R_{x,y,z} | \psi \rangle = \psi(x, y, z)\}] \qquad \textbf{(12.1.4.13)}$$

```
> Bra(psi) . P[R] . Ket(phi);
```

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \overline{\psi(x, y, z)}\, \phi(x, y, z) \, dx\, dy\, dz \qquad \textbf{(12.1.4.14)}$$

> *Bra*(phi) . *P*[*R*] . *Ket*(psi);

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\overline{\phi(x,y,z)}\,\psi(x,y,z)\,dx\,dy\,dz \qquad \textbf{(12.1.4.15)}$$

This is a bracket rule for the scalar product of a state-vector of the discrete basis *u* and | psi ⟩.

> *%Bracket*(*Bra*(*u*, *n*), *Ket*(psi)) = psi(*n*);

$$\langle u_n \mid \psi \rangle = \psi(n) \qquad \textbf{(12.1.4.16)}$$

> *Setup*(%);

$$[bracketrules = \{\langle u \mid v \rangle = f(u,v), \langle R_{x,y,z} \mid \phi \rangle = \phi(x,y,z), \langle R_{x,y,z} \mid \psi \rangle = \psi(x,y,z), \langle u_n \mid \psi \rangle = \psi(n)\}] \qquad \textbf{(12.1.4.17)}$$

This rule for $\langle u_n \mid$ psi ⟩ permits projecting ⟨ psi | psi ⟩ onto the *u* basis, which is equivalent to inserting a projector between ⟨ psi | and | psi ⟩.

> *%Bracket*(psi, *P*[*u*], psi);

$$\left\langle \psi \left| \sum_{n=0}^{N-1} \mid u_n \rangle \langle u_n \mid \right| \psi \right\rangle \qquad \textbf{(12.1.4.18)}$$

> *value*(%)

$$\sum_{n=0}^{N-1} |\psi(n)|^2 \qquad \textbf{(12.1.4.19)}$$

A **Ket** can have different types of spaces associated with its quantum numbers. In the following example, **Kets** from a basis *B* have four quantum numbers, two of which, B_3 and B_4, are associated with continuous spaces, and the dimension of the space associated with each quantum number is different.

> *Setup*(*quantumcontinuousbasis* = {*B*[3], *B*[4]}, *quantumbasisdimension* = {*B*[1] = -1/2..1/2, *B*[2] = 0..*N*, *B*[3] = -*a*..*a*, *B*[4] = -*infinity*..*infinity*});

$$[quantumbasisdimension = \{R = \infty, u = N, B_1 = -\frac{1}{2}..\frac{1}{2}, B_2 = 0..N, B_3 = -a..a, B_4 = -\infty..\infty\}, quantumcontinuousbasis = \{R, B_3, B_4\}] \qquad \textbf{(12.1.4.20)}$$

This is the projector onto the basis *B*:

> *Projector*(*Ket*(*B*, *n*, *m*, *x*, *y*));

$$\int_{-\infty}^{\infty}\int_{-a}^{a}\left(\sum_{m=0}^{N}\sum_{n=0}^{1}\left|B_{n-\frac{1}{2},m,x,y}\right\rangle\left\langle B_{n-\frac{1}{2},m,x,y}\right|\right)dx\,dy \qquad \textbf{(12.1.4.21)}$$

Quantum operators, eigenvectors, eigenvalues and commutators

To indicate to the system that a letter represents a quantum operator, use the Setup command; this sets *B* as a quantum operator.

> *Setup*(*op* = *B*);

** Partial match of 'op' against keyword 'quantumoperators'*

$$[quantumoperators = \{B\}] \qquad \textbf{(12.1.5.1)}$$

Note that noncommutative objects are displayed in different colors. To change this color, see ? Setup.

Because B is now a quantum operator, $|B_{m,n,x,y}\rangle$ is an eigenvector of the four operators B_1, B_2, B_3, and B_4, with eigenvalues m, n, x, and y, respectively. For example,

```
> B[2] . Ket(B, m, n, x, y);
```

$$n\,|B_{m,n,x,y}\rangle \qquad \textbf{(12.1.5.2)}$$

Quantum operators can also be 3-D Euclidean vectors; for that purpose, you must load the Physics[Vectors] subpackage.

```
> with(Vectors);
```

$$[\&x, `+`, `.`, ChangeBasis, Component, Curl, DirectionalDiff, Divergence, Gradient, Identify, Laplacian, \nabla, Norm, Setup, diff] \qquad \textbf{(12.1.5.3)}$$

Set the vectors $\vec{L}$, $\vec{r}$, and $\vec{p}$ as quantum operators (note the use of the option *redo* to erase previous definitions of quantum operators).

```
> Setup(op = {L_, r_, p_, x, y, z, p_x, p_y, p_z, L_x, L_y, L_z}, redo);
```

$$\text{* Partial match of 'op' against keyword 'quantumoperators'}$$

$$\left[quantumoperators = \left\{\vec{L}, L_x, L_y, L_z, \vec{p}, p_x, p_y, p_z, \vec{r}, x, y, z\right\}\right] \qquad \textbf{(12.1.5.4)}$$

Define $\vec{L}$ as the angular momentum operator $\vec{L} = \vec{r} \times \vec{p}$, and set commutation rules for the components of $\vec{r}$ and $\vec{p}$.

```
> L_ := r_ &x p_;
```

$$\vec{L} := \vec{r} \times \vec{p} \qquad \textbf{(12.1.5.5)}$$

```
> r_ := x*_i + y*_j + z*_k;
```

$$\vec{r} := \hat{i}\,x + \hat{j}\,y + \hat{k}\,z \qquad \textbf{(12.1.5.6)}$$

```
> p_ := p_x*_i + p_y*_j + p_z*_k;
```

$$\vec{p} := \hat{i}\,p_x + \hat{j}\,p_y + \hat{k}\,p_z \qquad \textbf{(12.1.5.7)}$$

To enter the commutation rules between each component of $\vec{r}$ and $\vec{p}$ with each other, you can write these commutators and pass the whole set to **Setup**. When there are many, as in this case, it is more convenient to use a Matrix and an indexing function. Enter the core information as a procedure: C represents the Commutator of the **Components** of the vectors $\vec{a}$ and $\vec{b}$.

```
> C := (a_, i, b_, j) → %Commutator(Component(a_, i), Component(b_, j));
```

$$C := \left(\vec{a}, i, \vec{b}, j\right) \mapsto \left[(\vec{a})_i, (\vec{b})_j\right]_- \qquad \textbf{(12.1.5.8)}$$

So, given i and j from 1 to 3 identifying the components of $\vec{r}$ and $\vec{p}$, an algebra can be set as set as follows.

```
> algebra := (i,j) → (
      C(r_, i, p_, j) = I*KroneckerDelta[i,j],
      C(r_, i, r_, j) = 0,
      C(p_, i, p_, j) = 0);
```

$$algebra := (i,j) \mapsto \left(C(\vec{r}, i, \vec{p}, j) = \mathrm{I}\,\delta_{i,j}, C(\vec{r}, i, \vec{r}, j) = 0, C(\vec{p}, i, \vec{p}, j) = 0\right) \qquad \textbf{(12.1.5.9)}$$

Now all of the commutators between each component of $\vec{r}$ and $\vec{p}$ can be constructed with one

call to Matrix.

```
> Matrix(3, 3, algebra);
```

$$\left[\left[\left([x,p_x]_- = \mathrm{I},\ [x,x]_- = 0,\ [p_x,p_x]_- = 0\right),\left([x,p_y]_- = 0,\ [x,y]_- = 0,\ [p_x,p_y]_- = 0\right),\left([x,p_z]_- = 0,\ [x,z]_- = 0,\ [p_x,p_z]_- = 0\right)\right],\right. \tag{12.1.5.10}$$
$$\left[\left([y,p_x]_- = 0,\ [y,x]_- = 0,\ [p_y,p_x]_- = 0\right),\left([y,p_y]_- = \mathrm{I},\ [y,y]_- = 0,\ [p_y,p_y]_- = 0\right),\left([y,p_z]_- = 0,\ [y,z]_- = 0,\ [p_y,p_z]_- = 0\right)\right],$$
$$\left.\left[\left([z,p_x]_- = 0,\ [z,x]_- = 0,\ [p_z,p_x]_- = 0\right),\left([z,p_y]_- = 0,\ [z,y]_- = 0,\ [p_z,p_y]_- = 0\right),\left([z,p_z]_- = \mathrm{I},\ [z,z]_- = 0,\ [p_z,p_z]_- = 0\right)\right]\right]$$

Pass this Matrix to Setup to set the algebra rules.

```
> Setup(%);
```

$$\left[algebrarules = \left\{[p_y,p_x]_- = 0,\ [p_z,p_x]_- = 0,\ [p_z,p_y]_- = 0,\ [x,p_x]_- = \mathrm{I},\ [x,p_y]_- = 0,\ [x,p_z]_- = 0,\ [y,p_x]_- = 0,\ [y,p_y]_- = \mathrm{I},\ [y,p_z]_- = 0,\ [y,x]_- = 0,\ [z,p_x]_- = 0,\ [z,p_y]_- = 0,\ [z,p_z]_- = \mathrm{I},\ [z,x]_- = 0,\ [z,y]_- = 0\right\}\right] \tag{12.1.5.11}$$

Set, for instance, the values of L_x, L_y, and L_z, the components of $\vec{L}$.

```
> L_x := _i . L_;
```

$$L_x := p_z\,y - p_y\,z \tag{12.1.5.12}$$

```
> L_y := _j . L_;
```

$$L_y := p_x\,z - p_z\,x \tag{12.1.5.13}$$

```
> L_z := _k . L_;
```

$$L_z := p_y\,x - p_x\,y \tag{12.1.5.14}$$

Verify the commutator algebra for these components of $\vec{L}$.

```
> Commutator(L_x, L_y) = I L_z;
```

$$-\mathrm{I}\,p_x\,y + \mathrm{I}\,p_y\,x = \mathrm{I}\left(p_y\,x - p_x\,y\right) \tag{12.1.5.15}$$

```
> Commutator(L_z, L_x) = I L_y;
```

$$-\mathrm{I}\,p_z\,x + \mathrm{I}\,p_x\,z = \mathrm{I}\left(p_x\,z - p_z\,x\right) \tag{12.1.5.16}$$

```
> Commutator(L_y, L_z) = I L_x;
```

$$-\mathrm{I}\,p_y\,z + \mathrm{I}\,p_z\,y = \mathrm{I}\left(p_z\,y - p_y\,z\right) \tag{12.1.5.17}$$

The three equations above are identically true.

Annihilation and Creation operators

Other operators frequently used in different contexts are the Annihilation and Creation operators: they augment or diminish the value of a quantum number by one. These operators

are suitable, for instance, for working with multi-particle vector states; in that context the quantum numbers are called occupation numbers.

This constructs a pair of annihilation/creation operators acting on the basis A involving only one quantum number.

> $am := Annihilation(A)$;

$$am := a-$$ **(12.1.5.18)**

> $ap := Creation(A)$;

$$ap := a+$$ **(12.1.5.19)**

Annihilation and **Creation** operators act on **Kets** belonging to discrete bases and assume that the "lower" state happens when the quantum number is equal to zero (frequently called "vacuum": a ket with occupation number equal to zero represents a state with "no particles").

> $K := Ket(A, n)$

$$K := |A_n\rangle$$ **(12.1.5.20)**

> $am \,.\, K$;

$$\sqrt{n}\,|A_{n-1}\rangle$$ **(12.1.5.21)**

> $am \,.\, \%$;

$$\sqrt{n}\,\sqrt{n-1}\,|A_{n-2}\rangle$$ **(12.1.5.22)**

> $am^k \,.\, K$ assuming $k :: nonnegint$, $n :: nonnegint$

$$\sqrt{(n-k+1)_k}\,|A_{n-k}\rangle$$ **(12.1.5.23)**

> $am^k \,.\, K$ assuming $nonnegint$

$$\sqrt{(n-k+1)_k}\,|A_{n-k}\rangle$$ **(12.1.5.24)**

> $convert($**(12.1.5.24)**, factorial$)$

$$\sqrt{\frac{n!}{(n-k)!}}\,|A_{n-k}\rangle$$ **(12.1.5.25)**

> $ap \,.\, K$;

$$\sqrt{n+1}\,|A_{n+1}\rangle$$ **(12.1.5.26)**

> $ap \,.\, \%$;

$$\sqrt{n+1}\,\sqrt{n+2}\,|A_{n+2}\rangle$$ **(12.1.5.27)**

> $ap^k \,.\, K$ assuming $nonnegint$

$$\sqrt{\frac{(n+k)!}{n!}}\,|A_{n+k}\rangle$$ **(12.1.5.28)**

The Commutator of the operators $a-$ and $a+$ are automatically set when these operators are constructed, and satisfy

> $(\%Commutator = Commutator)(am, ap)$;

$$[a-, a+]_- = 1$$ **(12.1.5.29)**

To indicate that the Kets of a basis are fermionic, use an anticommutative variable to label the

basis. To set the prefix identifier of anticommutative variables use the Setup command.

```
> Setup(anticommutativeprefix = Theta);
```

$$[anticommutativeprefix = \{\Theta, \_\lambda\}] \quad \textbf{(12.1.5.30)}$$

```
> type(Theta, anticommutative);
```

$$true \quad \textbf{(12.1.5.31)}$$

```
> Ket(Theta);
```

$$|\Theta\rangle \quad \textbf{(12.1.5.32)}$$

Construct **Annihilation** and **Creation** operators acting on this basis; use the option *notation = explicit* so that the basis and the quantum numbers onto which these operators act are explicit.

```
> Am := Annihilation(Theta, notation = explicit);
```

$$Am := a-_{\Theta_1} \quad \textbf{(12.1.5.33)}$$

```
> Ap := Creation(Theta, notation = explicit);
```

$$Ap := a+_{\Theta_1} \quad \textbf{(12.1.5.34)}$$

The AntiCommutator of these operators satisfy

```
> (%AntiCommutator = AntiCommutator)(Am, Ap);
```

$$\left[a-_{\Theta_1}, a+_{\Theta_1}\right]_+ = 1 \quad \textbf{(12.1.5.35)}$$

According to Pauli's exclusion principle, only one fermionic particle can be in a given state, so starting from the vacuum,

```
> Ket(Theta, 0);
```

$$|\Theta_0\rangle \quad \textbf{(12.1.5.36)}$$

```
> Ap . %;
```

$$|\Theta_1\rangle \quad \textbf{(12.1.5.37)}$$

```
> Ap . %;
```

$$0 \quad \textbf{(12.1.5.38)}$$

```
> Am . %%;
```

$$|\Theta_0\rangle \quad \textbf{(12.1.5.39)}$$

And as is always the case, the annihilation operator acting on the vacuum returns zero

```
> Am . %;
```

$$0 \quad \textbf{(12.1.5.40)}$$

From where: powers of anninilation/creation operators of fermionic types are equal to zero

```
> [Ap^2, Am^2]
```

$$\left[a+_{\Theta_1}{}^2, a-_{\Theta_1}{}^2\right] \quad \textbf{(12.1.5.41)}$$

```
> Simplify(%)
```

$$[0, 0] \quad \textbf{(12.1.5.42)}$$

For fermionic operators, thus, the occupation number operator is idempotent

```
> N := Ap Am
```

(12.1.5.43)

$$N := a+_{\Theta_1}\, a-_{\Theta_1} \tag{12.1.5.43}$$

> $N\,N\,N = N$

$$a+_{\Theta_1}\, a-_{\Theta_1}\, a+_{\Theta_1}\, a-_{\Theta_1}\, a+_{\Theta_1}\, a-_{\Theta_1} = a+_{\Theta_1}\, a-_{\Theta_1} \tag{12.1.5.44}$$

> $Simplify(\%)$

$$a+_{\Theta_1}\, a-_{\Theta_1} = a+_{\Theta_1}\, a-_{\Theta_1} \tag{12.1.5.45}$$

>

Exercises

1. Show that the commutator relation $[Q, P]_- = \mathrm{I}\,\hbar$ implies $\frac{\hbar^2}{4} \le \langle \mathrm{Delta}(P)^2 \rangle \langle \mathrm{Delta}(Q)^2 \rangle$

Solution

> $restart; with(Physics)$:

Consider two conjugate observables Q, P, and the corresponding Hermitian operators satisfying $[Q, P]_- = \mathrm{I}\,\hbar$.

> $macro(h = `\hbar`)$:

> $Setup(Hermitian = \{Q, P\}, \%Commutator(Q, P) = I * h)$;

* *Partial match of 'Hermitian' against keyword 'hermitianoperators'*

$$[algebrarules = \{[Q, P]_- = \mathrm{I}\,\hbar\}, hermitianoperators = \{P, Q\}] \tag{12.2.1.1}$$

Suppose now that the system where Q and P act is in some state | psi ⟩ normalized to 1, and set | psi ⟩ as the default state for computing Brackets.

> $Setup(\%Bracket(\mathrm{psi}, \mathrm{psi}) = 1, bracketbasis = \mathrm{psi})$;

$$[bracketbasis = \psi, bracketrules = \{\langle \psi \mid \psi \rangle = 1\}] \tag{12.2.1.2}$$

The mean values of the operators Q and P in the state | psi ⟩ are then given by:

> $Qm := Bracket(Q)$;

$$Qm := \langle Q \rangle \tag{12.2.1.3}$$

> $Pm := Bracket(P)$;

$$Pm := \langle P \rangle \tag{12.2.1.4}$$

Let's introduce another Hermitian operator, Delta, and denote $\mathrm{Delta}(Q)$ and $\mathrm{Delta}(P)$ the operators representing the observable deviations from these mean values by $\langle Q \rangle$ and $\langle P \rangle$.

> $Setup(her = \mathrm{Delta})$;

* *Partial match of 'her' against keyword 'hermitianoperators'*

$$[hermitianoperators = \{\Delta, P, Q\}] \tag{12.2.1.5}$$

> $DQ := \mathrm{Delta}(Q) = Q - Bracket(Q)$;

$$DQ := \Delta(Q) = Q - \langle Q \rangle \tag{12.2.1.6}$$

> $DP := \mathrm{Delta}(P) = P - Bracket(P)$;

$$DP := \Delta(P) = P - \langle P \rangle \tag{12.2.1.7}$$

The value of the Commutator between $\mathrm{Delta}(Q)$ and $\mathrm{Delta}(P)$ is a consequence of the value of

the Commutator between Q and P, and so it can be computed by rewriting the deviations in terms of Q and P.

```
> %Commutator(Delta(Q), Delta(P));
```

$$[\Delta(Q), \Delta(P)]_- \qquad \textbf{(12.2.1.8)}$$

```
> eval(%, {DQ, DP});
```

$$[Q - \langle Q \rangle, P - \langle P \rangle]_- \qquad \textbf{(12.2.1.9)}$$

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

$$I\,\hbar \qquad \textbf{(12.2.1.10)}$$

Track this result as an algebra rule, so that in what follows we compute directly with $\mathrm{Delta}(Q)$ and $\mathrm{Delta}(P)$.

```
> Setup(%%% = %);
```

$$[algebrarules = \{[Q, P]_- = I\,\hbar, [\Delta(Q), \Delta(P)]_- = I\,\hbar\}] \qquad \textbf{(12.2.1.11)}$$

To show now that $[Q, P]_- = I\,\hbar$ implies $\frac{\hbar^2}{4} \le \langle \mathrm{Delta}(P)^2 \rangle \langle \mathrm{Delta}(Q)^2 \rangle$, consider the action of these deviation operators $\mathrm{Delta}(Q)$ and $\mathrm{Delta}(P)$ on the state of the system $|\,\mathrm{psi}\,\rangle$, and construct with them a new Ket involving a real parameter lambda. When entering the following command, you will be asked whether it represents a function definition or a remember table assignment, choose remember table assignment (to perform these assignments with a function on the left-hand side without being asked questions enter first you can also enter *Typesetting*:-*Settings*('*functionassign* = *false*'))

```
> Ket(Psi, lambda) := (Delta(Q) + I*lambda*Delta(P)) . Ket(psi);
```

$$|\Psi_\lambda\rangle := \Delta(Q) \cdot |\psi\rangle + I\,\lambda\,(\Delta(P) \cdot |\psi\rangle) \qquad \textbf{(12.2.1.12)}$$

The square of the norm of $|\,\mathrm{Psi}_{\mathrm{lambda}}\,\rangle$, for lambda real, is

```
> Dagger(%) . % assuming lambda :: real;
```

$$\langle \Delta(P)^2 \rangle \lambda^2 - I \langle \Delta(P)\,\Delta(Q) \rangle \lambda + I\,\lambda \langle \Delta(Q)\,\Delta(P) \rangle + \langle \Delta(Q)^2 \rangle \qquad \textbf{(12.2.1.13)}$$

Simplify this norm, taking into account the commutator $[\mathrm{Delta}(Q), \mathrm{Delta}(P)]_- = I\,\hbar$, set in **(6.2.12)**

```
> Simplify(%);
```

$$\langle \Delta(P)^2 \rangle \lambda^2 - \lambda\,\hbar + \langle \Delta(Q)^2 \rangle \qquad \textbf{(12.2.1.14)}$$

This is a polynomial in lambda of second degree; its discriminant is negative or zero.

```
> discrim(%, lambda) ≤ 0;
```

$$\hbar^2 - 4 \langle \Delta(P)^2 \rangle \langle \Delta(Q)^2 \rangle \le 0 \qquad \textbf{(12.2.1.15)}$$

isolating $\frac{\hbar^2}{4}$, we obtain the lower bound for $\langle \mathrm{Delta}(P)^2 \rangle \langle \mathrm{Delta}(Q)^2 \rangle$.

```
> isolate(%, h^2)/4;
```

$$\frac{\hbar^2}{4} \le \langle \Delta(P)^2 \rangle \langle \Delta(Q)^2 \rangle \qquad \textbf{(12.2.1.16)}$$

Note that this result is a consequence of $[\mathrm{Delta}(Q), \mathrm{Delta}(P)]_- = I\,\hbar$, which in turn is a consequence of $[Q, P]_- = I\,\hbar$, so that Q and P too satisfy $\frac{\hbar^2}{4} \le \langle P^2 \rangle \langle Q^2 \rangle$, and in fact

the product of *any* two conjugate Hermitian operators, as well as of the root-mean square deviations of them, satisfy this inequality.

>

2. Let $\vec{L} = \vec{r} \times \vec{p}$. Show that if

$$\left[(\vec{r})_i, (\vec{p})_j\right]_- = \mathrm{I}\,\mathrm{delta}_{i,j}, \quad \left[(\vec{r})_i, (\vec{r})_j\right]_- = 0, \quad \left[(\vec{p})_i, (\vec{p})_j\right]_- = 0,$$

then

$$\left[\|\vec{L}\|^2, L_i\right]_- = 0 \quad \text{and} \quad \left[L_i, L_j\right]_- = \mathrm{I}\,\mathrm{epsilon}_{i,j,k} L_k$$

Solution