Tour of EQC functionality

Jan Rheinländer

March 18, 2012

Contents

1 Numeric evalution of equations

EQC adds the power of a spreadsheet to Latex. It is possible to write down equations and obtain the numerical value of variables:

```
$$\eq{x = 3}$$
$$\eq{y = 4}$$
$$\eq{z = \sqrt{x^2 + y^2}} = \val{z}$$
```

$$x = 3$$
$$y = 4$$
$$z = \sqrt{x^2 + y^2} = 5$$

If you only want to do a numerical calculation in your text, using the **val** keyword is sufficient.

```
The square root of two is \val{\sqrt{2}}.
```

The square root of two is 1.414.

Equations can be assigned a label to reuse them later on:

```
\begin{equation}
\eq[eq:important]{a = x + 2}
\end{equation}
\centerline{\emph{Equation \ref{eq:important} defines the value of a
to be $\printeq{"eq:important"} = \val{a}$.}}
```

$$a = x + 2 \tag{1}$$

Equation 1 defines the value of a to be $a = x + 2 = 5$.

If you want to use mathematical or physical constants in your equations, define them with the **constant** keyword. The file mathconstants.tex contains definitions for the most common constants π and e.

```
$$\constant{\Pi = 3.141}$$
$$\eq{r = 3}$$
$$\eq{d = 2\Pi r} = \val{d}$$
```

$$\Pi = 3.141$$
$$r = 3$$
$$d = 2\Pi r = 18.85$$

Equations can take an optional list of options, for example, `\eq[label=eq:x; eqraw=false]{x = y^2}`. The options will apply only to this specific equation. Some of the available options are:

units = {unit; unit; ...} Tell EQC what units it should use in the output.

precision = integer The precision (total number of digits) with which floating point numbers are printed.

fixeddigits = true|false You can decide whether you want to round all numbers to a fixed number of digits (e.g., 12345 rounded to a precision of three would be 12300) or whether to display a fixed number of digits after the decimal marker.

lowsclimit = real It is possible to control which numbers are printed in scientific notation (e.g., $1.23 \cdot 10^5$). This options give the smallest number that is still printed normally.

highsclimit = real This options give the largest number that is still printed normally.

eqalign = none|eqnarray|ams This option controls equation formatting. `none` prints no ampersands, `eqnarray` prints ampersands on both sides of the equals sign (`&=&`), and `ams` prints an ampersand before the equals sign (`&=`). By default, EQC automatically adds ampersands to your equations depending on the environment they appear in.

eqchain = true|false Omits the left hand side of an equation if it is identical to the left hand side of the equation directly preceding it in an `eqnarray` or AMS environment.

eqraw = true|false Print equations as you typed them (if possible), or with EQC formatting.

The keyword **eqcoptions** can be used to set these options globally.

2 Working with physical quantities

EQC adds the class Unit to the functionality of the GiNaC library. This means that you can use units in a natural way inside equations:

```
\clearequations
\eqcoptions{units = {\mm}}
$$\eq{x = \unit{3}{\mm}}$$
$$\eq{y = \unit{4}{\mm}}$$
$$\eq{z = x + y} = \quantity{z}$$
```

$$x = 3\ \mathrm{mm}$$

$$y = 4\ \mathrm{mm}$$

$$z = x + y = 7\ \mathrm{mm}$$

Upon startup, EQC knows only the base SI units. To get full support for all units defined in the SIunits.sty package, include units.tex at the beginning of your document. New units can easily be defined using the **defunit** keyword. For example:

```
\defunit[’’]{\inch}{2.54\mm}
\eqcoptions{units = {\inch}}
$$z = x + y = \quantity{z}$$
```

$$z = x + y = 2.756\ ''$$

The option `units` tells EQC what units it should use in the output. If no units are given, EQC will use the SI base units. The units can also be given as an optional argument to the **val** keyword (and will only apply to that specific value):

```
\defunit[’’]{\inch}{2.54\mm}
\eqcoptions{units = {\inch}}
$$z = x + y = \quantity{z} = \quantity[units = {\mm}]{z}$$
```

$$z = x + y = 2.756\ '' = 7\ \text{mm}$$

A short form also exists, with the syntax `\quantity[\mm]{z}`.

Note that there are four different keywords that can be used for finding the value of a variable:

numval This prints a numeric value, or prints an error if the variable does not have one.

units Prints only the units of the variable.

quantity Prints a numeric value plus units (if there are any).

val Prints any kind of symbolic expression

3 User defined functions

EQC extends GiNaC by offering user-defined functions which can be created at runtime in the Latex file. A function is declared using the **function** keyword. A definition for the function may be given with **deffunc**, but this is not strictly necessary. Of course, only functions with definitions can be evaluated!

Functions can be given “hints” when they are defined using the syntax `\function[hintlist]...`. Hints enable EQC to handle and print functions better. For example, the hint `trig` will result in the function being printed as $\sin^2 x$ instead of $(\sin(x))^2$.

Consider the function

```
\function{f}{x}
$$\deffunc{f}{ax^2 + bx + c}$$
```

$$f = ax^2 + bx + c$$

```
With $\eq{a = 4}$, $\eq{b = -2}$ and $\eq{c = 7}$ we can compute $f(x)$ for $x = 3$:
$$f(x) = \val{f(x)}$$
$$f(3) = \val{f(3)} = \numval{\val{f(3)}}$$
```

With $a = 4$, $b = -2$ and $c = 7$ we can compute $f(x)$ for $x = 3$:

$$f(x) = 7 + 4\,x^2 - 2\,x$$

$$f(3) = c + 9\,a + 3\,b = 37$$

Note what effect the **val** statement has on the different parameters. To both expand the function and substitute the values of known variables, a double **val** statement is necessary.

4 Asking for values of variables

Most documents will be of the type that define a number of equations and then try to calculate numeric values for the variables used in these equations. There are several methods of doing this efficienctly:

- The most straight-forward way is to define an equation and then other equations that give the known values of variables. The **val** statement can then be used to calculated the values of unkown variables.

```
\begin{eqnarray*}
    \eq{x = 3y + 4}\\
    \eq{y = 5}\\
    x &=& \numval{x}
  \end{eqnarray*}
```

$$\begin{aligned} x &= 3y + 4 \\ y &= 5 \\ x &= 19 \end{aligned}$$

  This method becomes awkward if you want to ask for the value of x again, using a different value for y. You would have to delete the equation defining y and create a new one.

- If you need the value of a variable for many different parameters, the best way is to define a function.

```
\clearequations
\begin{eqnarray*}
  \function{x}{y}%
  \deffunc{x}{3y+4}\\
  x &=& \val{x(5)}\\
  x &=& \val{x(7)}\\
\end{eqnarray*}
```

$$\begin{aligned} x &= 3y+4 \\ x &= 19 \\ x &= 25 \end{aligned}$$

The only drawback is that you cannot easily substitute the definition $x = 3y + 4$ into other equations. You would have to do something like `\eqsubst{z = 3x}{x = \val{x}}`.

- The last way is to use the `with`-form of the **\val** statement.

```
\clearequations%
\begin{eqnarray*}
  \eq{x = 3y + 4}\\
  x &=& \numvalwith{x}{y = 5}\\
  x &=& \numvalwith{x}{y = 7}\\
\end{eqnarray*}
```

$$\begin{aligned} x &= 3y+4 \\ x &= 19 \\ x &= 25 \end{aligned}$$

All the assignments given as the second argument are defined as equations, the search for the value is performed, and then the temporary equations are deleted again. Therefore, you can find even values of variables that are indirectly defined:

```
\clearequations
\begin{eqnarray*}
  \eq{x = 3y + 4}\\
  \eq{z = 4x}\\
  z &=& \numvalwith{z}{y = 5}\\
\end{eqnarray*}
```

$$\begin{aligned} x &= 3y+4 \\ z &= 4x \\ z &= 76 \end{aligned}$$

5 Symbolic computations

EQC offers the possibility of symbolic manipulation of equations. You can use it to add, subtract, multiply or divide an equation with an expression. Also, it is possible to substitute an expression with another expression, and to differentiate an expression.

5.1 Example: Solving a quadratic equation

Consider the equation

```
\clearequations
\function{f}{x}
$$\deffunc{f}{ax^2 + bx + c} = 0$$
```

$$f = ax^2 + bx + c = 0$$

What values of x will solve this equation?

We need to bring the equation into the form $x^2 + 2Ax + A^2 = B$, which can then be written as $(x + A)^2 = B$ and solved.

```
$$\eqdiv{\val{f(x)} = 0}{a}, a \ne 0$$
$$\eqadd{"prev"}{(1/2 b/a)^2 - c/a}$$
$$\eqsimpf{"prev"}{expand}$$
```

$$\frac{1}{a}\left(c + x\,b + x^2\,a\right) = 0, a \neq 0$$

$$\frac{1}{4}\frac{b^2}{a^2} + \frac{1}{a}\left(c + x\,b + x^2\,a\right) - \frac{c}{a} = \frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}$$

$$x^2 + \frac{1}{4}\frac{b^2}{a^2} + \frac{x\,b}{a} = \frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}$$

The special equation label `"prev"` is a shortcut to access the last equation processed by EQC. With **\eqsimpf** it is possible to do a variety of simplifications on the equation:

expand Fully expands all expressions, including function arguments.

expandf Only expand function definition, not arguments.

eval Numerically evaluate the equation as far as possible.

normal Normalize the equation, that is, force all terms to have a common denominator (see description of GiNaC's `normal()` method for details).

collect-common Collect common factors from all terms of the equation (see description of GiNaC's `collect_common_factors()` method for details).

unsafe Do unsafe simplifications, for example, $\sqrt{x^2} = x$ and $\arctan \tan x = x$. Note that the opposite, $\tan \arctan x$, is not unsafe and thus is done automatically (by GiNaC).

As can be seen, $A = \frac{1}{2}\frac{b}{a}$ and $B = (\frac{1}{2}\frac{b}{a})^2 - \frac{c}{a}$. The equation can now be written as:

```
$$\eq{\left(x + \frac{1}{2}\frac{b}{a}\right)^2 = \rhs{"prev"}}$$
```

$$\frac{1}{4}\left(2\,x + \frac{b}{a}\right)^2 = \frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}$$

Applying the square root to both sides yields the two solutions:

```
$$\eqpow{"prev"}{1/2}$$
$$\eqsimpf[eq:poss1]{"prev"}{unsafe}$$
$$\mbox{or}\quad\eq[eq:poss2]{\lhs{"prev"} = -\rhs{"prev"}}$$
```

$$\frac{1}{2}\sqrt{\left(2\,x + \frac{b}{a}\right)^2} = \sqrt{\frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}}$$

$$x + \frac{b}{2\,a} = \sqrt{\frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}}$$

$$\mbox{or}\quad x + \frac{b}{2\,a} = -\sqrt{\frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}}$$

The two possible values for x are therefore:

```
\eqsubst*[eq:temp1]{"eq:poss1"}{x = x_1}
\eqsubst*[eq:temp2]{"eq:poss2"}{x = x_2}

$$\eqsub[eq:sol1]{"eq:temp1"}{\frac{b}{2a}}$$
$$\eqsub[eq:sol2]{"eq:temp2"}{\frac{b}{2a}}$$
```

$$x_1 = \sqrt{\frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}} - \frac{b}{2\,a}$$

$$x_2 = -\frac{b}{2\,a} - \sqrt{\frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}}$$

A faster way to achieve the same result is by using **eqsolve**.

```
$$\eqsubst{\eqsolve{f(x) = 0}{x}{1}}{x = x_1}$$
$$\eqsubst{\eqsolve{f(x) = 0}{x}{2}}{x = x_2}$$
```

$$x_1 = \sqrt{\frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}} - \frac{b}{2\,a}$$
$$x_2 = -\frac{b}{2\,a} - \sqrt{\frac{1}{4}\frac{b^2}{a^2} - \frac{c}{a}}$$

With, for example, $a = 3$, $b = 4$ and $c = -5$ the possible values for x are:

```
$$x_1 = \numval[3]{x_1}$$
$$x_2 = \numval[3]{x_2}$$
```

$$x_1 = 0.786$$
$$x_2 = -2.12$$

The `3` option to **\numval** sets the precision to three digits, it is equivalent to writing `\numval[precision=3]{x_1}`.
When applying $f(x)$ on these values of x, the result is (almost) zero:

```
$$f(x_1) = \numval{f(x_1)}$$
$$f(x_2) = \numval{f(x_2)}$$
```

$$f(x_1) = -1.776 \cdot 10^{-15}$$
$$f(x_2) = 0$$

5.2 Example: Finding the extrema of a function

To find the extrema of $f(x)$, we set the first derivative to zero and solve for x:

```
\deleq{"eq:a";"eq:b";"eq:c"}
$$\eq[eq:f]{f(x) = \val{f(x)}}$$
$$\eqdiff{"eq:f"}{x} = 0$$
\eqsub*{\rhs{"prev"} = 0}{b}%
\eqsubst*{"prev"}{x = x_{extr}}%
$$\eqdiv[eq:xextr]{"prev"}{2a}$$
```

$$f(x) = c + x\,b + x^2\,a$$
$$f'(x) = b + 2\,x\,a = 0$$
$$x_{extr} = -\frac{b}{2\,a}$$

Depending on the sign of the second derivate $f''(x) = 2\,a$, this will be a minimum or a maximum. The value of the extremum is

```
$$\eqsubstc[eq:yextr]{y_{extr} = \rhs{"eq:f"}}{x = x_{extr}; "eq:xextr"}$$
```

$$y_{extr} = c - \frac{1}{4}\frac{b^2}{a}$$

or with the values used in the previous section $a = 3$, $b = 4$ and $c = -5$:

```
$$x_{extr} = \numval{x_{extr}}\quad y_{extr} = \numval{y_{extr}}$$
```

$$x_{extr} = -0.6667 \quad y_{extr} = -6.333$$

5.3 Library of substititions

To make working with equations easier, there is a library of mainly trigonometric substitutions which can be included with `\input substitutions.tex`. For example, consider this equation:

```
\clearequations
\input ../examples/substitutions.tex
$$\eq[eq:r1]{\frac{r_1}{\sin(\pi - \frac{\theta}{2})} = \frac{r_2}{\sin(\frac{\pi}{2} - \
$$\eqsubst{"prev"}{"lib:trig:sina+pi/2"; "lib:trig:sina+pi"}$$
$$\eqsubst{"prev"}{"lib:trig:sin-a"; "lib:trig:cos-a"}$$
```

$$\frac{r_1}{\sin(\pi - \frac{\theta}{2})} = \frac{r_2}{\sin(\frac{\pi}{2} - \frac{\theta}{2})}$$
$$-\frac{r_1}{\sin\left(-\frac{\theta}{2}\right)} = \frac{r_2}{\cos\left(-\frac{\theta}{2}\right)}$$
$$\frac{r_1}{\sin\left(\frac{\theta}{2}\right)} = \frac{r_2}{\cos\left(\frac{\theta}{2}\right)}$$

This equation can now be solved to give an expression for $\frac{\theta}{2}$:

```
$$\eqsubst{"prev" * \sin{\theta/2} / r_2}{\eqrev{"lib:trig:tansincos"}}$$
$$\eqrev{\eqsimpf{\eqfunc{"prev"}{\arctan}}{unsafe} * 2}$$
```

$$\frac{r_1}{r_2} = \tan\left(\frac{\theta}{2}\right)$$
$$\theta = 2\ \arctan\left(\frac{r_1}{r_2}\right)$$

The keyword **eqrev** swaps left and right hand side of the equation. **eqfunc** applies a function to both sides.

6 Matrices

Vectors and matrices can be created with the following input:

```
\begin{align*}
  \matrix{v_1}%
  \matrix{v_2}%
  \matrix{M_1}%
  \eq[eqraw=false]{v_1 = {x; y; z}}\\
  \eq[eqraw=false]{v_2 = \transpose{x; y; z}}\\
  \eq[eqraw=false]{M_1 = {{x_1; y_1; z_1};{x_2; y_2; z_2};{x_3; y_3; z_3}}}
\end{align*}
```

$$v_1 = \begin{pmatrix} x & y & z \end{pmatrix}$$
$$v_2 = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$
$$M_1 = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix}$$

It is very important to indicate to EQC that a variable contains a matrix (because of the non-commutativity of products). This is done with the **matrix** keyword. Note that currently it is not possible to create a "standing" vector with `\eq{v_2 = {{x};{y};{z}}`. Instead, use the **transpose** function as shown in the example.

The value of matrix elements can be found with the function `\mindex`:

```
\begin{align*}
  \eq{m = \mindex{M_1; b; 2}}\\
  \eq{b = 3}\\
  m_{32} &= \val{m}
\end{align*}
```

$$m_{32} = M_1[b, 2]$$
$$b = 3$$
$$m_{32} = y_3$$

You can use the `\wild` function as the index to access a whole row or column.

EQC supports the following operations with matrices and vectors:

- Addition and subtraction
- Multiplication
- Exponentiation
- Transposition with the **transpose**

EQC will terminate with an error if the matrices or vectors are not compatible with one another.

7 Graphs

EQC has some support for creating graphs with the package pstricks. As an example, we will compare the graph of a function with its Taylor series expansion.

The definition of the Taylor series expansion is

$$T_n f(x, x_0) = \sum_{i=0}^{n} f^{(i)}(x_0) \cdot \frac{(x - x_0)^i}{i!}$$

Applying this to $\arctan(x)$ around the point $x_0 = 0.7$, we have

```
\function{f}{x}%
\begin{align}
  \eq[eq:f0]{f(x) = \arctan x}\tag*{}\\
  \eqdiff[eq:f1]{"prev"}{x}\tag*{}\\
  \eqdiff[eq:f2]{"prev"}{x}\tag*{}
\end{align}
```

$$\begin{aligned} f(x) &= \arctan x \\ f'(x) &= \frac{1}{1+x^2} \\ f''(x) &= -2\frac{x}{\left(1+x^2\right)^2} \end{aligned}$$

The values of the function at x_0 are

```
\begin{align*}
  \eqsubstc{"eq:f0"}{x = x_0; "eq:x_0"}\\
  \eqsubstc{"eq:f1"}{x = x_0; "eq:x_0"}\\
  \eqsubstc{"eq:f2"}{x = x_0; "eq:x_0"}
\end{align*}
```

$$\begin{aligned} f(0.7) &= 0.6107 \\ f'(0.7) &= 0.6711 \\ f''(0.7) &= -0.6306 \end{aligned}$$

The first parts of the expansion are

```
\begin{align}
  \eq{\arctan(x) =
    \rhs{\eqsubst{"eq:f0"}{x = x_0}} +
    \rhs{\eqsubst{"eq:f1"}{x = x_0}} \frac{x - x_0}{1!} +
    \rhs{\eqsubst{"eq:f2"}{x = x_0}} \frac{(x - x_0)^2}{2!}
    }\notag\\
  \eqsubst[eq:series]{"prev"}{"eq:x_0"}\tag*{}\\
  \eqsimpf{"prev"}{expand}
\end{align}
```

$$\begin{aligned} \arctan x &= \arctan x_0 - \frac{x_0 - x}{1+x_0^2} - \frac{x_0\left(x_0 - x\right)^2}{\left(1+x_0^2\right)^2} \\ &= 0.1409 + 0.6711\,x - 0.3153\left(x - 0.7\right)^2 \\ &= 1.113\,x - 0.3153\,x^2 - 0.01357 \qquad (2) \end{aligned}$$

The same result can be obtained more easily with the **\tseries** keyword:

```
  \arctan x = \val{\tseries{\arctan x}{x = 0.7}{3}}
```

$$\arctan x = 1.113\,x - 0.3153\,x^2 - 0.01357$$

The two functions are compared in figure 1.

```
\begin{figure}[!htb]
  \begin{center}
    \psset{xunit=40mm, yunit=25mm}
    \mmakepspicture{-1}{-1.6}{2}{1.2}{0.5}{0.2}{$x$}{$\arctan(x)$}{%IMPORTANT COMMENT!
      \printvector{\val{\eqeval{y = \arctan x}{x = -1:2:0.1}}}%
    \psset{linecolor=red}%
    \pscurve{-}\printvector{\val{\eqeval{"eq:series"}{x = -1:2:0.1}}}}
  \end{center}
  \caption{Comparison of $\arctan x$ with its Taylor series expansion\label{fig:func}}
\end{figure}
```

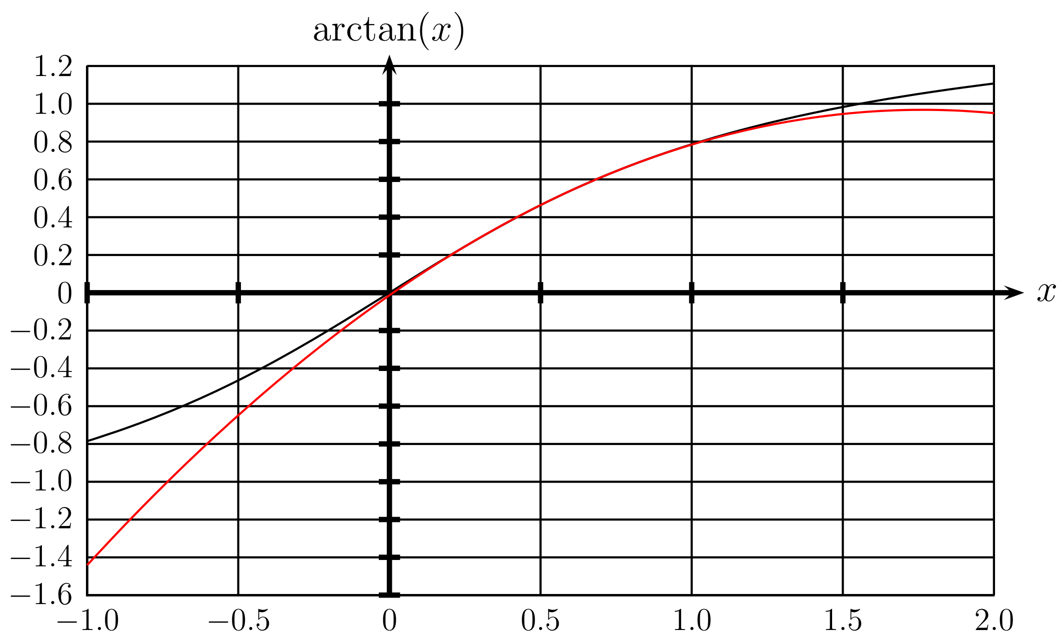

Figure 1: Comparison of $\arctan x$ with its Taylor series expansion

The arguments to **mmakepspicture** are:

- Negative x-axis value, then negative y-axis value
- Positive x-axis value, then positive y-axis value
- x-axis step value, then y-axis step value (for the axis labels)
- x and y axis names
- An array of $(x; y)$ values to plot

The last is generated by the keyword **\printvector**, which converts the output of **\eqeval** into a format recognized by pstricks. **\eqeval** evaluates an equation for all the values of the independent variable given as the second parameter. In the example, $-1 : 2 : 0.1$ creates a vector of numbers from -1 to 2 in increments of 0.1. The result is a matrix with the values of the independent variable in the first column and the evaluation results in the second column:

```
\eq{r = \eqeval{"eq:series"}{x = -1:2:0.5}}
```

$$r = \begin{pmatrix} -1 & -1.441 \\ -0.5 & -0.6487 \\ 0 & -0.01357 \\ 0.5 & 0.4639 \\ 1 & 0.7837 \\ 1.5 & 0.9458 \\ 2 & 0.9504 \end{pmatrix}$$

There is also a simplified version **\makepspicture** for drawing a graph in the first quadrant only:

- Location of origin in the format (x0, y0)
- Location of lower left-hand corner in the format (x1, y1)
- Positive x-axis value, then positive y-axis value
- x-axis step value, then y-axis step value (for the axis labels)
- x and y axis names
- An array of $(x; y)$ values to plot

Of course, if **\makepspicture** and **\mmakepspicture** do not suit your purposes, you can use the standard pstricks macros instead or create your own shortcuts.

8 Utility functions

If you want to make a fresh start in the middle of a document, use **\clearequations**. This will delete all previously defined equations except library equations, but keep the constants and any functions defined with the hint `lib`.

9 Pitfalls

9.1 Unexpected newlines

If you use EQC commands like `\eqmul*` that produce no Latex output inside an equationarray environment, follow them by a comment to avoid insertion of a newline in the output file. Equationarray environments may not contain newlines. For example:

```
\begin{eqnarray}
\eq{x & = & a + b}
\eqsub*{"prev"}{b}
\eqrev{"prev"}
\end{eqnarray}
```

will yield the following output:

```
\begin{eqnarray}
x&=&a+b

a&=&x-b
\end{eqnarray}
```

9.2 Multiple possible values for equations

Consider the following equation:

$$u = x + y \sin\varphi$$

Now we want to find the value x_1 which will make u become zero.

```
\begin{equation}
\eqsubst[eq:usubst]{"prev"}{x = x_1} = 0
\end{equation}
```

$$u = x_1 + y\ \sin\varphi = 0 \tag{3}$$

and find x_1 to be:

```
\eqsubst*{"prev"}{u = 0}
\eqsub*{"prev"}{\sin\varphi y}
$$\eqrev{"prev"}$$
```

$$x_1 = -y\ \sin\varphi$$

But what happens if we now ask for the value of u?

```
u = \val{u}
```

$$u = x_1 + y\ \sin\varphi$$

EQC prints a warning that there are multiple possible values for u (i.e., $x + y \sin\varphi$ and $x_1 + y \sin\varphi$) and then chooses the last value (which might or might not be what we wanted). There are two ways to avoid this:

1. Do both substitutions at the same time: `\eqsubst{"prev"}{x = x_1; u = 0}`.
2. Introduce a temporary for u: `\eqsubst{"prev"}{x = x_1; u = u_{temp}}`.

9.3 Multiple substitutions

If you do multiple substitutions at the same time, the order in which the substitutions are done is not defined. By using the keyword **eqsubstc**, substitutions are done consecutively in the order they are listed. For example, compare:

```
$$\eqsubst{x = 3y}{y = 4z; z = 3}$$
$$\eqsubstc{x = 3y}{y = 4z; z = 3}$$
```

$$x = 12\,z$$

$$x = 36$$

Only use **eqsubstc** when order is important because it is less efficient than **eqsubst**.

9.4 Output of numbers

Not all combinations of `\precision`, `\precision_type` and `\scientific_limits` make sense. For example, you might set the lower scientific limit to 0.001 so that numbers like 0.005 will be printed without exponent. If you now set `precision_type` to `fixed_marker` and precision to 2, numbers smaller than 0.01 will be printed as zero!

9.5 Library equations

Note that library equations are not automatically used for finding values of variables. They are considered to be purely for reference purposes. An example:

```
\eq[lib:myeq]{x = 7y}
\eq{y = 2}
x = \val{x}
```

This will *not* return 14 as the value of x. If you want to use a library equation for finding values, you need to "activate" it. There are two different possibilities:

```
\eq{"lib:myeq"}
\eqsubst{"lib:myeq"}{x = x_1}
```

The last case is really what the library equations are there for: They need to be adapted to special purposes by substituting custom variables into them.

9.6 Pstricks headaches

pstricks can give you headaches when the function that is to be plotted has x or y values of less than 1.0, for example, $\sin x$:

```
\begin{figure}[!htb]
\begin{center}
\pspicture(\val{-180\degree},-1)(\val{180\degree},1)
  \psgrid(\val{-180\degree},-1)(\val{180\degree},1)
  \psaxes[linewidth=2pt, labels=none]{->}(0,0)(\val{-180\degree},-1)(\val{180\degree},1)
  \pscurve{-}\printvector{\val{
      \eqeval{y = \sin x}{x = -180\degree:180\degree:30\degree}}}
\endpspicture
\end{center}
\caption{Function $y = \sin x$, naïve version}
\end{figure}
```

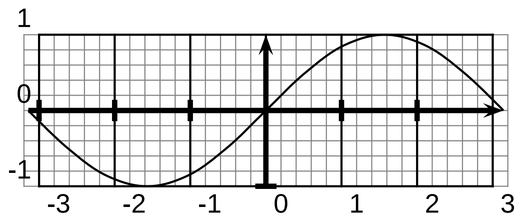

Figure 2: Function $y = \sin x$, naïve version

Note that some **val** statements are required to evaluate the units to floating point numbers. The method above has two major drawbacks: The x-axis is scaled in radians, not in degrees, and the numbering of the y-axis is only in integer numbers.

The non-naïve version of the diagram is:

```
\begin{figure}[!htb]
\begin{center}
\psset{xunit=0.35mm, yunit=40mm}
\pspicture(-190,-1.05)(200,1.1)
  \psaxes[linewidth=2pt, ticks=none, labels=none]{->}(0,0)(-190,-1.05)(200,1.1)
  \psaxes[Dx=30, ticks=x, labels=x, ticksize=40mm]{-}(0,0)(-180,-1)(180,1)
  \psaxes[Dy=0.2, ticks=y, labels=y, ticksize=54mm]{-}(0,0)(-180,-1.05)(180,1.1)
  \pscurve{-}\printvector{\val{\eqeval{
        \eqsubst{y = \sin x}{x = x \degree}}{x = -180:180:30}}}
\endpspicture
\end{center}
\caption{Function $y = \sin x$}
\end{figure}
```

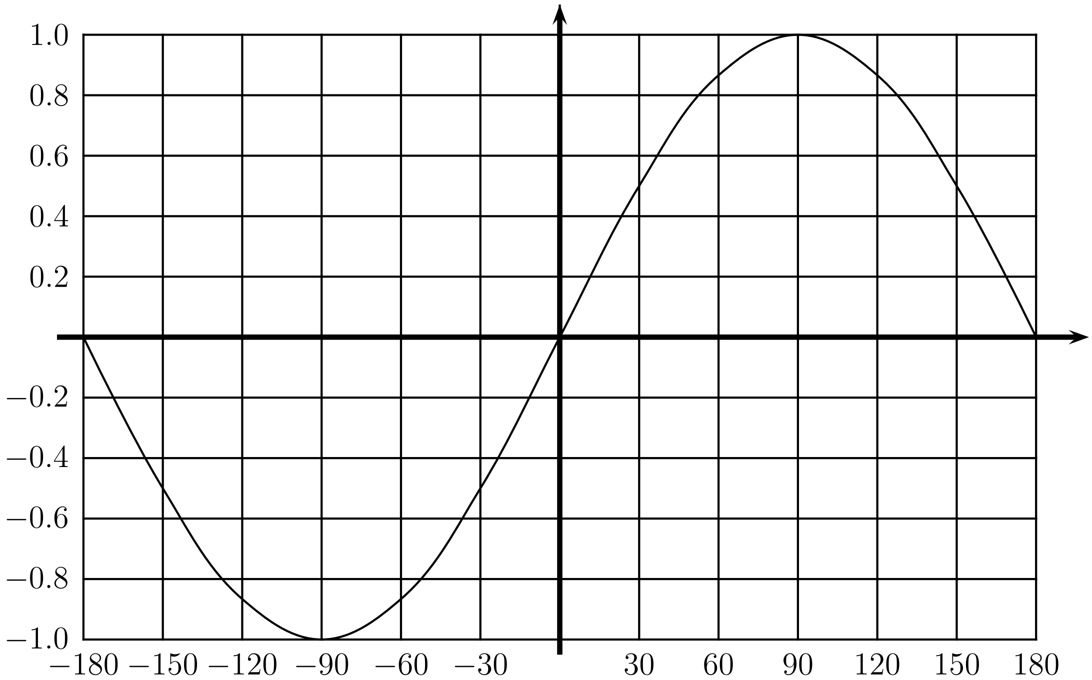

Figure 3: Function $y = \sin x$