

# Smath Documentation

T. Stumpf-Fétizon

December 13, 2014

`smath` is a collection of  $\text{\LaTeX}$  macros that automates and centralizes most of the formatting in math mode. Instead, you declare a variable to belong to some class of mathematical objects (e.g. a vector) and the package defines the appropriate formatting for that class. The overall goal is to let you focus on the meaning of your writing, just as outside of math mode.

## 1 Use Cases

My beef with  $\text{\LaTeX}$  is that it requires me to do a lot of manual formatting which makes the code crowded and unreadable. Moreover, direct formatting violates the basic principle of “what you see is what you mean”. Finally, I think that math mode should work like CSS in that there should be some document-wide classes that define how a certain mathematical object should be formatted. The class adds semantic meaning to the markup while taking charge of the formatting. Notice that while these issues may seem obvious and their solution trivial, I have not encountered a standard solution yet. Hence this package.

### 1.1 Standard Formatting

Consider a mathematical object that is given a standard formatting, say a vector. These are usually printed bold ( $\mathbf{x}$ ). For roman letters, we use the `\mathbf` command. This does not work for greek letters, so we need to use the `\boldsymbol` command. Dealing with different commands to format the same object is cumbersome. Furthermore, the commands have little semantic meaning as other objects (say matrices) may use the same formatting. The obvious solution is to define a macro that is applied to every vector and that takes care of the different formatting for different alphabets. To stick with the vector case, you can use the `\nvec` command to mark any roman or greek letter as a vector in your source code. It will then be boldened automatically. If you prefer to format all your vectors by using an arrow, you can do so by changing the macro.

## 1.2 Delimiter Scaling

Another issue concerns the use of delimiters and parentheses for operator functions (e.g. the exponential function). In complicated expressions, we need to use the `left` and `right` commands to scale parentheses so that they “hug” fractions and inner parentheses correctly. This often piles up to a ton of scaling commands that completely obscure the meaning of the markup. Nonetheless, notice that a large proportion of braces and parentheses are used to declare functions and operators. Adequate macros can automate the formatting of operator names and the scaling of the parentheses. For example, the `op` command prints the name of the operator in straight letters (as customary) and encloses the argument in correctly scaled parentheses. When using the macros, you should only have to deal with delimiters when you are actually grouping terms.

## 1.3 Complex Expressions

Some areas of mathematics, especially statistics, contain a large amount of long and reoccurring expressions (think density functions). The package provides macros that let you print some of those expressions for variable arguments with a single command.

## 2 Coding Examples

This section contains some examples of how the package is supposed to make your life easier.

### 2.1 The Trace of the Hat Matrix

The `op` command has two parameters which are the name of the operator and its argument. It prints the name in straight letters and encloses the argument in scaling parentheses. As an application, consider the trace of the hat matrix:

$$\mathrm{tr}(\mathbf{H}) = \mathrm{tr}\left(\Phi(\Phi^T\Phi)^{-1}\Phi^T\right) \quad (1)$$

Now consider two equivalent ways of coding the equation above:

Listing 1: Coding the trace of the hat matrix

```
1 % standard latex code
2 \begin{equation}
3   \mathrm{tr}(\mathbf{H}) = \mathrm{tr} \left( \boldsymbol{\Phi} \right.
4     \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \left. \right)
5 \end{equation}
6 % code with smath macros
7 \begin{equation}
```

---

```

8   \op{tr}{\mat{H}} = \op{tr}{\mat{\Phi} (\mat{\Phi}\T \mat{\Phi}
    \Phi)}^{-1} \mat{\Phi}\T}
9   \end{equation}

```

---

Using a few widely applicable macros, we halved the length of the markup and made its mathematical meaning far more obvious.

## 2.2 The Multivariate Gaussian

Have a look at the density of the multivariate gaussian distribution.

$$p(\mathbf{x}, \boldsymbol{\mu}, \mathbf{Q}) = (2\pi)^{-\frac{D}{2}} \det(\mathbf{Q})^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{Q} (\mathbf{x} - \boldsymbol{\mu})\right) \quad (2)$$

It may take minutes for an inexperienced typist to code that equation in L<sup>A</sup>T<sub>E</sub>X. Have a look at two equivalent versions of the underlying L<sup>A</sup>T<sub>E</sub>Xcode:

---

Listing 2: Coding the density of the multivariate gaussian distribtion

---

```

1  % standard latex code
2  \begin{equation}
3    p(\mathbf{x}, \boldsymbol{\mu}, \mathbf{Q}) = (2\pi)^{-\frac{D}{2}} \mathrm{det}(\mathbf{Q})^{\frac{1}{2}} \exp \left(
      -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{Q} (\mathbf{x} - \boldsymbol{\mu}) \right)
4  \end{equation}
5
6  % code with smath macros
7  \begin{equation}
8    \fun{p}{\nvec{x}, \nvec{\mu}, \mat{Q}} = (2\pi)^{-\frac{D}{2}} \op{det}{\mat{Q}}^{\frac{1}{2}} \op{exp}{-\frac{1}{2}
      (\nvec{x} - \nvec{\mu})\T \mat{Q} (\nvec{x} - \nvec{\mu})}
9  \end{equation}

```

---

It actually took me 2 minutes to write the first version. The macros cut out 25% of the code and again improve clarity. You would obtain even greater gains if you needed to scale more braces.

Some expression occur so often in similar form that it may be more efficient to print them using macros. To stick with the previous example, you can print it using the following snippet:

---

Listing 3: Coding the density with a macro

---

```

1  \begin{equation}
2    \fun{p}{\nvec{x}, \nvec{\mu}, \mat{Q}} = \dmgauss{\nvec{x}}{\nvec{\mu}}{\mat{Q}}{D}
3  \end{equation}

```

---

It gives you some flexibility, but formatting might not be adequate for some arguments. Nevertheless, there are many circumstances in which you can save a lot of time by using these canned macros.

### 3 Using `smath`

Place `smath.sty` in the same folder as your `.tex` file and add `\usepackage{smath}` to your preamble. The package requires `mathtools` to be loaded. You can now use any macro defined in `smath.sty`. Consult the file for macro definitions. You can easily extend the package by adding your own macros to the file. Consult your favorite L<sup>A</sup>T<sub>E</sub>X documentation on how to define macros.