The engsymbols package*

Fábio Fortkamp
fabio@fabiofortkamp.com

March 8, 2016

1 Introduction

This document describes the engsymbols package, a collection of macros to facilitate the writing of common engineering symbols.

The following packages are prerequisites:

- bm
- amsmath
- esdiff

This package follows the conventions specified by ISO standards of typesetting mathematics [1]. The main points of this convention are:

- variables are always writen in italic, *and also* in bold for vector and matrices
- in other words, vectors don't have arrows above it
- subscripts and superscripts are written in roman upright font when they do not indicate a variable per se; e.g. when they refer to a physicial object, a fluid fase, a state etc

engsymbols is actually just a collection of commands I, as a Ph.D. student in Mechanical Engineering, find useful, and I hope others can find it to. There aren't any special design principles, except that:

- Command names are not short, but are descriptive. The user can always define an alias, and many editors can parse the packages and offer tab completion.
- These macros are not intended to be the absolute true way of scientific writing. As I said, I try to follow some standards and observe what is commonly used.

Please notice that the user should refer to other references such as papers and textbooks to get the meaning of the symbols I describe here.

*This document corresponds to engsymbols v0.2, dated 2016/03/08.

2 Available commands

2.1 Special symbols

First we have some basic symbols, commands that take no arguments.

`\volume` This `\volume` macro typesets a special symbol $\mathcal{V}$ to indicate the total volume of something. The more logical V is avoided, mainly in fluid mechanics texts, because it's used for velocity. I chose the calligraphic V because the often used $\forall$ is really ugly and is intended to use with logical propositions ("for all x ...").

`\hheat` `\hmass` The symbols $\hbar$ (`\hheat`) and $\hbar_{\mathrm{m}}$ (`\hmass`) are used in some textbooks to indicate the convection heat (or mass) transfer coeficient and to avoid confusion with enthalpy h. I'm aware that $\hbar$ is also used for the reduced Planck constant, but I don't think it's used in engineering literature. I also know that the distinction between an enthalpy value and a heat transfer coefficient value can be inferred by the context, but in a equation with many terms, including terms with enthalpy and heat transfer rates, this can make things clearer. Also notice the m subscript in roman fonts; it's a coefficient for mass transfer, and not a coefficient for a variable m.

`\diffusivitybinary` I deal with mixtures a lot, so there is `\diffusivitybinary` $\mathcal{D}_{12}$, which is used to indicate the mass diffusivity of a binary mixture.

`\universalgasconstant` Finally, there is a command `\universalgasconstant` to write R_{u}, the universal gas constant. The u index (also in roman font) marks this as a constant; each fluid has its own "gas constant", defined as $R = \frac{R_{\mathrm{u}}}{M}$, where M is the molar mass; hence for a mixture, this gas constant varies with the composition, and of course different fluids have different R. R_{u} is universal for all fluids.

2.2 Auxiliary commands

`\ped` `\ap` These macros are defined and explained in [1], and produce a subscript and superscript in roman font; this is probably the most confusing command. `h\ped{L}` produces h_{L}, the enthalpy of the liquid phase, while `e_L` produces e_L, for example, the error in a *variable* L. These commands can also be writen in text mode, like `2\ap{nd}` to write 2^{nd}.

2.3 Vector commands

`\nvector` `\nmatrix` The command `\nvector{`$\langle arg \rangle$`}` typesets `{`$\langle arg \rangle$`}` in vector form, like $\boldsymbol{x}$ and $\boldsymbol{\phi}$. There is also a matrix command. The "n" indicates notation, to avoid defining something as general as `\vector`.

`\univector` `\normalvector` For typesetting unit vectors, you can type `\unitvector{\alpha}` to get a unit vector in the direction of α as $\hat{\boldsymbol{e}}_\alpha$. To get the commonly-used normal vector $\hat{\boldsymbol{n}}$, you can type `\normalvector`.

`\vectornorm` `\divergent` `\curl` The other commands are used in vector operations. The vector norm $\|a\|$ is written as `\vectornorm{\nvector{a}}`, the divergent $\nabla \cdot \boldsymbol{V}$ as `\divergent{\nvector{V}}`, and the vector curl $\nabla \times \boldsymbol{V}$ as `\curl{\nvector{V}}`. The gradient of a scalar ∇T (`\gradient{T}`) transforms a scalar into a vector (which is why I included it into this group of "vector operations"), and the laplacian $\nabla^2 x$, with many applications in differential equations, is defined as `\laplacian{x} = \divergent{\gradient{x}}` (try to understand this definition and to imagine how it would be typeset).

`\laplacian` `\divergentpar` `\gradientpar` `\laplacianpar`

The `\laplacian`, `\divergent` and `\gradient` also have `par` variants to automatically add parentheses; for example, `\divergentpar{\rho \nvector{V}}` is the term $\nabla \cdot (\rho \boldsymbol{V})$ widely used in fluid mechanics.

`\divergentv` `\curlv`

Finally, for the divergent (and curl), there are commands `\divergentv{q}`, `\curlv{q}` that automatically wraps the argument into vector form, as in $\nabla \cdot \boldsymbol{q}$.

2.4 Other operations

`\rate` `\flux`

The command `\rate{m}` produces $\dot{m}$, which can be understood as a "mass flow rate". This is just an alias for `\dot`, and is actually harder to type, but it's more meaningful. Similarly, there is `\flux{q}` to write the heat flux q''.

`\average`

The macro `\average{f}` results in $\overline{f}$. I've seen some definitions that shortens the bar above the argument, but I think this length is just fine.

`\diffpar` `\diffppar`

Sometimes I want to write the derivative of a product; I really like the `\diff` command from the esdiff package, but $\frac{\mathrm{d}\rho V}{\mathrm{d}x}$ is a little ugly for my taste. I prefer something like $\frac{\mathrm{d}}{\mathrm{d}x}(\rho V)$: the numerator in the derivative is empty, and the argument, the function to be derived, is written right after the fraction with parentheses around it (this expression was written as `\diffpar{\rho V}{x}`). There is also the `\diffppar{\rho V}{x}` variant, resulting in $\frac{\partial}{\partial x}(\rho V)$.

3 Implementation

3.1 Basic operations

`\ped` `\ap`

These macros by [1] typesets the argument in math roman font, to indicate a object. Italic subscripts should be used only to refer to another variables, for example, c_P is the specific heat obtained by mantaining the pressure, a physical parameter, fixes. By contrast, h_L (produced by `h\ped{L}`) is the liquid enthalpy; liquid is not a variable. The command `\ap{`⟨*index*⟩`}` does the same to superscripts, like T^I for the interface temperature.

```
\newcommand{\ped}[1]{\ensuremath{_{\mathrm{#1}}}}
\newcommand{\ap}[1]{\ensuremath{^{\mathrm{#1}}}}
```

`\nvector` `\nmatrix`

We define vector and matrix commands according to ISO standards: bold italic for vectors ($\boldsymbol{x}$) and matrices ($\boldsymbol{A}$). The "n" in names stands for "notation". This requires the `bm` package.

```
\newcommand{\nvector}[1]{\bm{#1}}
\newcommand{\nmatrix}[1]{\bm{#1}}
```

3.2 Special individual symbols

`\volume` This macro produces a calligraphic V to indicate volume, as $\mathcal{V}$. This is usually done to avoid confusion with velocity.

```
\newcommand{\volume}{\mathcal{V}}
```

`\diffd` This macro produces the differential d operator, as in $\mathrm{d}x$. The definition is fairly complex beacuse it tries to do an optimal spacing, and is described by [1].

```
\newcommand{\diffd}{\@ifnextchar^{\DIfF}{\DIfF^{}}}
\def\DIfF^#1{%
  \mathop{\mathrm{\mathstrut d}}%
      \nolimits^{#1}\gobblespace}
\def\gobblespace{%
  \futurelet\diffarg\opspace}
\def\opspace{%
  \let\DiffSpace\!%
  \ifx\diffarg(%
      \let\DiffSpace\relax
  \else
      \ifx\diffarg[%
          \let\DiffSpace\relax
      \else
          \ifx\diffarg\{%
              \let\DiffSpace\relax
          \fi\fi\fi\DiffSpace}
```

`\hheat` `\hmass` These macros produces a "crossed" h as in $\hbar$. This is done in some texts to denote the convection heat transfer coefficient and differentiate it from enthalpy h. This is actually just an alias to the existing command `\hbar`, to give a more meaningful name. There is also `\hmass` to produce $\hbar_{\mathrm{m}}$, used to indicate a mass transfer coefficient.

```
\newcommand{\hheat}{\hbar}
\newcommand{\hmass}{\hbar\ped{m}}
```

`\universalgasconstant` A simple command to produce R_{u}

```
\newcommand{\universalgasconstant}{R\ped{u}}
```

`\diffusivitybinary` This is a shorthand for the diffusivity of a binary mixture, $\mathcal{D}_{12}$.

```
\newcommand{\diffusivitybinary}{\mathcal{D}_{12}}
```

`\normalvector` This produces $\hat{\boldsymbol{n}}$.

```
\newcommand{\normalvector}{\hat{\nvector{n}}}
```

3.3 Common operations

`\average` This command puts a line above the argument (like $\overline{x}$), a notation widely used to indicate some type of average.

```
\newcommand{\average}[1]{\overline{#1}}
```

\rate

This macro denotes the rate of something, like $\dot{m}$ for a mass flow rate.

```
\newcommand{\rate}[1]{\dot{#1}}
```

\flux

Produces q''.

```
\newcommand{\flux}[1]{{#1}''}
```

\divergent
\divergentpar
\divergentv

These two macros produce the diverget of a vector $\nabla \cdot \boldsymbol{q}$. The `par` variant automatically adds parentheses, useful for multiple arguments like $\nabla \cdot (\rho \boldsymbol{V})$ (produced with `\divergentpar{\rho \nvector}`). The `\divergentv` command automatically converts the argument to a vector

```
\newcommand{\divergent}[1]{\nabla \cdot #1}
\newcommand{\divergentv}[1]{\divergent{\nvector{#1}}}
\newcommand{\divergentpar}[1]{\divergent{\left( #1 \right)}}
```

\curl
\curlv

These two macros produce the diverget of a vector $\nabla \times \boldsymbol{q}$. The `\curlv` command automatically converts the argument to a vector

```
\newcommand{\curl}[1]{\nabla\times{#1}}
\newcommand{\curlv}[1]{\curl{\nvector{#1}}}
```

\gradient
\gradientpar

Gradient of a scalar ∇T. The `par` variant introduces parentheses (e.g. $\nabla\left(\frac{\rho_1}{\rho}\right)$.

```
\newcommand{\gradient}[1]{\nabla {#1}}
\newcommand{\gradientpar}[1]{\gradient{\left( {#1} \right)}}
```

\laplacian
\laplacianpar

The laplacian of a scalar x is defined as $\nabla^2 x = \nabla \cdot \nabla x$. One could also use $\nabla^2 \left(\rho c_p T\right)$.

```
\newcommand{\laplacian}[1]{\nabla^2 #1}
\newcommand{\laplacianpar}[1]{\laplacian{\left( #1 \right)}}
```

vectornorm

Produces the norm of a vector, like $\left\lVert \boldsymbol{V} \right\rVert$.

```
\newcommand{\vectornorm}[1]{\left\lVert #1 \right\rVert}
```

\unitvector

Typesetts the unit vector in a given direction.

```
\newcommand{\unitvector}[1]{\hat{\nvector{e}}_{#1}}
```

\diffpar
\diffppar

This produces something like $\frac{\mathrm{d}}{\mathrm{d}x}\left(\rho V\right)$, building on the `\diff` command from the esdiff package. I find it really useful for printing derivatives when the function to be derived is a product of variables. Compare with $\frac{\mathrm{d}\rho V}{\mathrm{d}x}$. Notice the parentheses are automatically added. There is also the `\diffppar` for partial derivatives.

```
\newcommand{\diffpar}[2]{\diff{}{#2} \left( #1 \right)}
\newcommand{\diffppar}[2]{\diffp{}{#2} \left( #1 \right)}
```

References

[1] Claudio Beccari. Typesetting mathematics for science and technology according to iso 31/xi. *TUGboat*, 18(1):39–48, 1997.