The `smart-eqn` Package:
Automatic Math Symbol Styling

Ziyue "Alan" Xiang

ziyue.alan.xiang@gmail.com

CTAN: http://www.ctan.org/pkg/smart-eqn

VC: https://github.com/xziyue/smart-eqn

August 6, 2021

Contents

1 Introduction

In LaTeX typesetting, it is usually the case that one needs to use different variants of a math symbol to clarify the meanings. For example, in linear algebra literature, it is common to use boldfaced symbols to represent vectors, and use normal symbols to represent scalars, which makes equations like $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ easier to understand. However, applying these variants by typing `\mathbf`, `\mathrm` commands manually can be daunting. The smart-eqn package aims to provide an automatic and customizable approach for math symbol styling, which eliminates the need to enter style commands repeatedly.

2 Usage

2.1 Style Configuration

> **\smesetsym**{⟨*style csname*⟩}{⟨*symbols*⟩}

The ⟨*symbols*⟩ given in the form of comma separated list will be styled using ⟨*style csname*⟩. In Example 1, the four symbols A, v, Q, and Λ will be styled with the command \symbfup from the unicode-math package, which produces boldfaced symbols. If traditional LaTeX is used, one can use the \mathbf command instead.

> **\smeclearsym**

Clear the style configuration associated with all symbols.

2.2 Inline Math

In traditional LaTeX, inline math is typeset with $...$. In smart-eqn, we use @...@ instead. For inline math demonstrations, see Examples 2, 3, 4, 5.

> **\makeatmath**

The command changes the category code of @ so that it can be used as the inline math environment. To revert the change, one can use \makeatletter or \makeatother.

2.3 Display Math

In order to use display math environments, we need to declare smart math environments using \smenewenv. The names of new math environments will be prefixed with "sme". The way of passing arguments to the math environment (e.g., alignat) will also be slightly different. For display math demonstrations, see Examples 6, 7, 8, 9, 10.

> **\smenewenv**{⟨*env name*⟩}

Declares a "smart" math environment based on the traditional math environment provided in ⟨*env name*⟩. The name of the new math environment will be prefixed with "sme". For example, if one calls \smenewenv{align}, then the smart math environment smealign will be available.

Passing arguments   The arguments for math environments needed to be enclosed in square brackets. See Example 10.

2.4 Raw Math Content

> **\smeraw**{⟨*content*⟩}

The ⟨*content*⟩ of \smeraw will not be styled. This can be useful when one needs to style symbols manually in smart-eqn environments. See Examples 5, 9.

3   Examples

The following code snippet is executed before running the examples:

```
\makeatmath
\smenewenv{align}
\smenewenv{gather}
\smenewenv{alignat}
```

EXAMPLE 1 (Setting styles for symbols)

```
\smesetsym{symbfup}{A, v, Q, \Lambda}
```

EXAMPLE 2 (Inline math example 1)

```
\smesetsym{symbfup}{A, v}
The eigenvalue @\lambda@ and eigenvector @v@ of matrix @A@ satisfy @Av=\
    lambda v@.
```

The eigenvalue λ and eigenvector $\mathbf{v}$ of matrix $\mathbf{A}$ satisfy $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$.

EXAMPLE 3 (Inline math example 2)

```
\smesetsym{symbfup}{Q, \Lambda, I}
A symmetric matrix @A@ can be orthogonally diagonalized, that is, @A=Q\
    Lambda Q^T@, where @Q^TQ=I@ and @\Lambda@ is diagonal.
```

A symmetric matrix $\mathbf{A}$ can be orthogonally diagonalized, that is, $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{T}$, where $\mathbf{Q}^{T}\mathbf{Q} = \mathbf{I}$ and $\mathbf{\Lambda}$ is diagonal.

EXAMPLE 4 (Inline math example 3)

```
\smesetsym{mathcal}{S}
For a set @S@, the power set of @S@ (denoted by @2^S@) is the set of all
    subsets of @S@.
```

For a set $\mathscr{S}$, the power set of $\mathscr{S}$ (denoted by $2^{\mathscr{S}}$) is the set of all subsets of $\mathscr{S}$.

EXAMPLE 5 (Inline math example 4)

```
I just want to juxtapose @S \smeraw{S} \smeraw{\symbfup{S}} \smeraw{\
    symbfit{S}} \smeraw{\mathrm{S}}@.
```

I just want to juxtapose $\mathscr{S}S\mathbf{S}\mathbfit{S}\mathrm{S}$.

EXAMPLE 6 (Display math example 1)

```
\begin{smealign}
Av_1&=\lambda_1 v_1\\
Av_2&=\lambda_2 v_2
\end{smealign}
```

$$\mathbf{A}\mathbf{v}_1 = \lambda_1\mathbf{v}_1 \tag{1}$$
$$\mathbf{A}\mathbf{v}_2 = \lambda_2\mathbf{v}_2 \tag{2}$$

EXAMPLE 7 (Display math example 2)

```
\begin{smegather}
k_1v_1 + k_2v_2 + \cdots + k_n v_n = \symbfup{0}\\
l_1v_1 + l_2v_2 + \cdots + l_{n-1} v_{n-1} = \symbfup{1}
\end{smegather}
```

$$k_1\mathbf{v}_1 + k_2\mathbf{v}_2 + \cdots + k_n\mathbf{v}_n = \mathbf{0} \tag{3}$$
$$l_1\mathbf{v}_1 + l_2\mathbf{v}_2 + \cdots + l_{n-1}\mathbf{v}_{n-1} = \mathbf{1} \tag{4}$$

EXAMPLE 8 (Display math example 3)

```
\begin{smealign}
e^{At} = I+\sum_{n=1}^{\infty} \frac{A^n t^n}{n!}
\end{smealign}
```

$$e^{\mathbf{A}t} = \mathbf{I} + \sum_{n=1}^{\infty} \frac{\mathbf{A}^n t^n}{n!} \tag{5}$$

EXAMPLE 9 (Display math example 4)

```
\begin{smealign}
S
\smeraw{S}
\smeraw{\symbfup{S}}
\smeraw{\symbfit{S}}
\smeraw{\mathrm{S}}
\end{smealign}
```

$$\mathscr{S}S\mathbf{S}S \tag{6}$$

```
\smesetsym{symbfup}{x,b}
\begin{smealignat}[{5}]
&&\phantom{\Rightarrow} && (c_1^2 + \cdots + c_n^2) \lambda_n^{-2} &\leq \
    lVert \delta x \rVert^2 &&\leq (c_1^2 + \cdots + c_n^2) \lambda_1
    ^{-2}\\
&&\Rightarrow &&\lVert \delta b \rVert^2\lambda_n^{-2} &\leq \lVert \delta
    x \rVert^2 &&\leq \lVert \delta b \rVert^2\\
&&\Rightarrow &&\lVert \delta b \rVert^2\lambda_n^{-2} &\leq \lVert \delta
    x \rVert^2 &&\leq \lVert \delta b \rVert^2
\end{smealignat}
```

$$(c_1^2 + \cdots + c_n^2)\lambda_n^{-2} \leq \|\delta\mathbf{x}\|^2 \leq (c_1^2 + \cdots + c_n^2)\lambda_1^{-2} \tag{7}$$

$$\Rightarrow \quad \|\delta\mathbf{b}\|^2\lambda_n^{-2} \leq \|\delta\mathbf{x}\|^2 \leq \|\delta\mathbf{b}\|^2 \tag{8}$$

$$\Rightarrow \quad \|\delta\mathbf{b}\|^2\lambda_n^{-2} \leq \|\delta\mathbf{x}\|^2 \leq \|\delta\mathbf{b}\|^2 \tag{9}$$

4 Implementation

```
1 \RequirePackage{fancyvrb}
2 \RequirePackage{expl3, xparse}
3
4 \makeatletter
5 \ExplSyntaxOn
6
7 \tl_new:N \g_sme_op_arg_tl
8 \ior_new:N \g_sme_tmpa_ior
9 \iow_new:N \g_sme_tmpa_iow
10 \prop_new:N \g_sme_symbols_prop
11 \clist_new:N \l_sme_tmpa_clist
12 \tl_new:N \l_sme_tmpa_tl
13 \tl_new:N \l_sme_tmpb_tl
14 \tl_new:N \l_sme_tmpc_tl
15 \tl_new:N \l_sme_cur_math_tl
```

\smeDefineVerbatimEnvironment

This is a modified version of \DefineVerbatimEnvironment from fancyvrb. In this function, \smeFV@Environment is used insetad of \FV@Environment.

```
16 \def\smeDefineVerbatimEnvironment#1#2#3{%
17   \@namedef{#1}{\smeFV@Environment{#3}{#2}}%
18   \@namedef{end#1}{\@nameuse{FVE@#2}}%
19 }
```

\smeFV@Environment

This is a modified version of \FV@Environment from fancyvrb. In this function, \smeFV@GetKeyValues is used instead of \FV@GetKeyValues.

```
20 \def\smeFV@Environment#1#2{%
21   \def\FV@KeyValues{#1}%
22   \catcode`\^^M=\active
23   \tl_gclear:N \g_sme_op_arg_tl % clear optional arguments from previous calls
24   \@ifnextchar[%
25     {\catcode`\^^M=5 \smeFV@GetKeyValues{\@nameuse{FVB@#2}}}%
26     {\catcode`\^^M=5 \@nameuse{FVB@#2}}}
```

5

`\smeFV@GetKeyValues`

This is a modified version of `FV@GetKeyValues` from `fancyvrb`. We directly save the optional parameters captured by `fancyvrb` to the variable `\g_sme_op_arg_tl`, which can be used later.

```
27 \def\smeFV@GetKeyValues#1[#2]{%
28   \tl_gset:Nn \g_sme_op_arg_tl {#2}#1}
```

`\sme_declare_math_env:n`

#1: math environment name

This function declares a "smart" math environment. It is based on the `VerbatimOut` environment from `fancyvrb`, which writes its content out to the file `\jobname-sme.vrb`. The content stored in the external file will be read back and processed.

```
29 \cs_set:Npn \sme_declare_math_env:n #1 {
30     \exp_args:Nc \def{sme#1}{\smeFV@Environment{}{sme#1}}
31
32     % this is a modified version of VerbatimOut environment which does not take any argument
33     \exp_args:Nc \def{FVB@sme#1}{%
34       \@bsphack
35       \begingroup
36         \FV@UseKeyValues
37         \FV@DefineWhiteSpace
38         \def\FV@Space{\space}%
39         \FV@DefineTabOut
40         \def\FV@ProcessLine{\immediate\write\FV@OutFile}%
41         \immediate\openout\FV@OutFile \jobname-sme.vrb\relax
42         \let\FV@FontScanPrep\relax
43 %% DG/SR modification begin - May. 18, 1998 (to avoid problems with ligatures)
44         \let\@noligs\relax
45 %% DG/SR modification end
46         \FV@Scan}
47
48     \exp_args:Nc\def{FVE@sme#1}{\immediate\closeout\FV@OutFile\endgroup\@esphack
49         % call the function to process the content when the environment ends
50         \sme_read_process_math:n {#1}
51     }
52
53     \smeDefineVerbatimEnvironment{sme#1}{sme#1}{}
54 }
55
56 % define the user function
57 \let\smenewenv\sme_declare_math_env:n
```

`\sme_read_process_math:n`

#1: math environment name

Read back and process the math content stored in the external file `\jobname-sme.vrb`. After processing, the new content will be stored in another file `\jobname-sme-in.vrb`. Eventually, the new content will be fed into LaTeX using `\input`.

```
58 \cs_set:Npn \sme_read_process_math:n #1 {
59     \ior_open:Nn \g_sme_tmpa_ior {\jobname-sme.vrb}
60     \iow_open:Nn \g_sme_tmpa_iow {\jobname-sme-in.vrb}
61
62     \iow_now:Nx \g_sme_tmpa_iow {\c_backslash_str begin{#1} \g_sme_op_arg_tl}
63     \ior_map_inline:Nn \g_sme_tmpa_ior {
64         \sme_process_math:n {##1}
65         \exp_args:NNV \iow_now:Nn \g_sme_tmpa_iow \l_sme_cur_math_tl
66     }
67     \iow_now:Nx \g_sme_tmpa_iow {\c_backslash_str end{#1}}
68     \iow_close:N \g_sme_tmpa_iow
69     \ior_close:N \g_sme_tmpa_ior
70     \input{\jobname-sme-in.vrb}
71 }
```

`\smesetsym`

stores the style information in `\g_sme_symbols_prop`

```
72 \newcommand{\smesetsym}[2]{
73     \clist_set:Nn \l_sme_tmpa_clist {#2}
74     \clist_map_inline:Nn \l_sme_tmpa_clist {
75         \prop_gput:Nnn \g_sme_symbols_prop {##1} {#1}
76     }
77 }
```

`\smeclearsym`

clear the style information stored in `\g_sme_symbols_prop`

```
78 \newcommand{\smeclearsym}{
79     \prop_gclear:N \g_sme_symbols_prop
80 }
```

`\smeraw`

used to apply custom styles in smart-eqn environments

```
81 \newcommand{\smeraw}[1]{#1}
```

`\__sme_grouped:n`

```
82 \cs_set:Npn \__sme_grouped:n #1 {
83     \exp_not:n {#1}
84 }
85 \cs_generate_variant:Nn \__sme_grouped:n {e}
86 \cs_generate_variant:Nn \__sme_grouped:n {V}
```

`\sme_construct_bm:Nn`

#1: result variable

#2: current token list

This function is the core of automatic math styling. It examines each token in ⟨*current token list*⟩ and determines if it needs special styling. The function runs recursively and stores the result in ⟨*result variable*⟩.

```
87 \cs_set:Npn \sme_construct_bm:Nn #1#2 {
88     \tl_if_head_is_group:nTF {#2} {
89         % if head is group, apply the algorithm recursively
90         %\tl_put_right:Nx #1 {{\tl_head:n {#2}}}
91         \tl_put_right:NV #1 \c_left_brace_str
92         \exp_args:NNx \sme_construct_bm:Nn #1 {\tl_head:n {#2}}
93         \tl_put_right:NV #1 \c_right_brace_str
94         \tl_if_empty:nF {#2} {\exp_args:NNx \sme_construct_bm:Nn #1 {\tl_tail:n {#2}}}
95     }{
96         \tl_if_head_is_space:nTF {#2} {
97             % ignore spaces
98             \exp_args:NNx \sme_construct_bm:Nn #1 {\tl_trim_spaces:n {#2}}
99         } {
100            \tl_if_head_eq_meaning:nNTF {#2} \smeraw {
101                % for \smeraw, ignore the next group
102                \tl_put_right:Nn #1 {\smeraw}
103                \tl_set:Nx \l_sme_tmpa_tl {\tl_tail:n {#2}}
104                \tl_put_right:Nx #1 {\tl_head:N \l_sme_tmpa_tl}
105                \tl_set:Nx \l_sme_tmpa_tl {\tl_tail:N \l_sme_tmpa_tl}
106                \exp_args:NNV \sme_construct_bm:Nn #1 \l_sme_tmpa_tl
107            } {
108                % common case
109                % extract head
110                \exp_args:NNx \prop_get:NnNTF \g_sme_symbols_prop {\tl_head:n {#2}} \l_sme_tmpc_tl
    {
```

7

```
111                    % this symbol needs to be styled
112                    \tl_set:No \l_sme_tmpa_tl {\csname\l_sme_tmpc_tl\endcsname}
113                    \tl_put_right:Nx \l_sme_tmpa_tl {{\__sme_grouped:e {\tl_head:n {#2}}}}
114                    \tl_put_right:Nx #1 {{\__sme_grouped:V \l_sme_tmpa_tl}}
115                } {
116                    % otherwise, use the original symbol
117                    \tl_put_right:Nx #1 {\tl_head:n {#2}}
118                }
119                \tl_if_empty:nF {#2} {\exp_args:NNx \sme_construct_bm:Nn #1 {\tl_tail:n {#2}}}
120            }
121        }
122    }
123 }
```

\sme_process_math:n

#1: inline content

used by inline math environment to process the content

```
124 \cs_set:Npn \sme_process_math:n #1 {
125     \tl_clear:N \l_sme_cur_math_tl
126     \sme_construct_bm:Nn \l_sme_cur_math_tl {#1}
127 }
```

\makeatmath

changes the catcode and definition of @ so that we can use it for math typesetting

```
128 \begingroup
129 \catcode`@=\active
130 \gdef\makeatmath{%    note the global \gdef
131   \catcode`@=\active
132   \def@##1@{
133     \sme_process_math:n{##1}
134     $\exp_args:NnV \tl_rescan:nn {} \l_sme_cur_math_tl$
135   }%
136 }
137 \endgroup
138 \makeatother
139 \ExplSyntaxOff
```