

Package `sobolev`*

F. Bosisio

E-mail: `fbosisio@bigfoot.com`

1997/11/14

Abstract

Documentation for the package `sobolev`.

1 Introduction

This package provides some commands which are useful when dealing with Sobolev spaces and their relatives.

In particular some commands are redefined, so care should be taken, especially when including this package in an already existent \LaTeX file.

The redefined commands are `\H` and `\L`. The effect of “`\H`” (which is a type of accent) can now be achieved by the command “`\Haccent`”, whilst the job of “`\L`” (i.e. print an “L” with a superimposed bar) is now done by the command “`\Lbar`”.

2 The options

Two options are available at the moment: `DivInBrackets` and `DivAsExponent`. They only affect the output of the “`\Hdiv`” command.

The first options (`DivInBrackets`, which is the default) makes `\Hdiv` behave like “`H(div;...)`”, while the second one (`DivAsExponent`) makes `\Hdiv` expand to “`H^{div}(...)`”.

*This is version 2.2, last revised 1997/11/14; documentation date 2005/04/09

3 The commands

Most of the subsequent space-generating commands have mandatory arguments to indicate the type of the space. Often this argument consists of a single digit: in this case it is *not* necessary to enclose it in brackets, since in L^AT_EX the names of commands consists of letters only, and so a digit following it is certainly an argument. This saves a lot of typing and is the only reason that makes these commans useful (if you always had to type the brackets, then it would have been simpler to type the expansion of the command than the command itself !). In other words, you can think as if several commands exist (like `\H`, `\H1`, `\H10`, etc.), the ones with the digit beeing a sort of abbreviation for the general one.

3.1 The H command

The `\H` command is used to generate the symbol of sobolev spaces. It takes a mandatory argument, which is used as a superscript, and an optional argument, which is used as a subscript.

As explained above, if the mandatory argument is a digit, it need not be enclosed in brackets. Moreover, if the optional argument is the digit “0”, it can be typed without the square brackets.

Here are some examples (whith the `\DefaultSet` set to its default value `\Omega`):

$$\begin{array}{ll} \code{\H2} & \implies H^2(\Omega) \\ \code{\H10} & \implies H_0^1(\Omega) \\ \code{\H1[\Gamma_D]} & \implies H_{\Gamma_D}^1(\Omega) \\ \code{\H{-1/2}} & \implies H^{-1/2}(\Omega) \end{array}$$

3.2 The Hdiv command

The `\Hdiv` command is used to generate the sobolev space called “H div”. It takes only an optional argument, which is used as a subscript and which need not to be surrounded by the square brackets if it is the digit “0”.

If the (default) option `DivInBrackets` is in effect, it differs from the command `\H` in that the word “div” is printed (in roman type) inside brackets, before the set. If, instead, the option `DivAsExponent` is active, then it is simply an abbreviation for `\H{\mathrm{div}}`.

Here are some examples:

| | | DivInBrackets | DivAsExponent |
|------------------------------|------------|--|---|
| <code>\Hdiv</code> | \implies | $H(\operatorname{div}; \Omega)$ | $H^{\operatorname{div}}(\Omega)$ |
| <code>\Hdiv0</code> | \implies | $H_0(\operatorname{div}; \Omega)$ | $H_0^{\operatorname{div}}(\Omega)$ |
| <code>\Hdiv[\Gamma_D]</code> | \implies | $H_{\Gamma_D}(\operatorname{div}; \Omega)$ | $H_{\Gamma_D}^{\operatorname{div}}(\Omega)$ |

3.3 The L command

The `\L` command is used to generate the symbol of Lebesgue-measurable functions. It has one argument which is the exponent of the L-space. Again, if this argument is a digit (or a single symbol, like “`\infty`”) the surrounding braces are optional. Like for the `\H` command, the output of `\DefaultSet` is appended.

| | | | |
|-------------------------|-----------------------|------------|--------------------|
| | <code>\L2</code> | \implies | $L^2(\Omega)$ |
| Here are some examples: | <code>\L{10}</code> | \implies | $L^{10}(\Omega)$ |
| | <code>\L\infty</code> | \implies | $L^\infty(\Omega)$ |

3.4 The W command

The `\W` command is completely analogous, except that it prints a “W” instead of an “L” and that it has two argument, both printed as a superscript, separated by a comma. It is used for the generalized Sobolev spaces.

| | | | |
|---------------------------------------|------------------------|------------|------------------------|
| | <code>\W{k}{p}</code> | \implies | $W^{k,p}(\Omega)$ |
| Here is an example of how it is used: | <code>\W1\infty</code> | \implies | $W^{1,\infty}(\Omega)$ |

3.5 The D command

The `\D` command is used in the theory of distributions: it prints the space of distributions over the `\DefaultSet` if followed by a prime symbol, or its dual space, otherwise.

| | | |
|------------------|------------|------------------------|
| <code>\D</code> | \implies | $\mathcal{D}(\Omega)$ |
| <code>\D'</code> | \implies | $\mathcal{D}'(\Omega)$ |

3.6 The Norm command

The `\Norm` command has a mandatory and an optional argument; it generates the norm of the mandatory argument, with the optional argument, if present, as a whole subscript, to denote the space within which the norm is taken.

| | | | |
|----------------|----------------------------|------------|---------------|
| Some examples: | <code>\Norm{f(x)}</code> | \implies | $\ f(x)\ $ |
| | <code>\Norm{g}[L^2]</code> | \implies | $\ g\ _{L^2}$ |

3.7 The SemiNorm command

The `\SemiNorm` command is completely analogous, but generates the semi-norm instead of the norm.

Some examples: $\backslash\text{SemiNorm}\{f(x)\} \implies |f(x)|$
 $\backslash\text{SemiNorm}\{g\}[H^1] \implies |g|_{H^1}$

3.8 The Scalar command

The `\Scalar` command has two arguments; a third optional argument (which is used as a whole subscript) may follow inside square brackets. The output consists of the two arguments separated by a comma and enclosed in a pair of adjustable-size brackets, with the optional argument placed as a subscript (to denote the space inside which the scalar product is taken).

Some examples: $\backslash\text{Scalar}\{f\}\{g\} \implies (f, g)$
 $\backslash\text{Scalar}\{u\}\{v\}[L^2] \implies (u, v)_{L^2}$

3.9 The Crochet command

The `\Crochet` command has two arguments; a third optional argument (which is used as a whole subscript) may follow inside square brackets. The output consists of the two arguments separated by a comma and enclosed in a pair of adjustable-size angular-parentheses, with the optional argument placed as a subscript (to denote the space inside which the duality is taken).

Some examples: $\backslash\text{Crochet}\{f\}\{g\} \implies \langle f, g \rangle$
 $\backslash\text{Crochet}\{u\}\{v\}[D] \implies \langle u, v \rangle_D$

4 Implementation

```

1 %%
2 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3 \ProvidesPackage{\FileName}[\filedate\space v\fileversion\space\filedescr]

```

The options set the flag for the `\Hdiv` command.

```

4 %%
5 \newif\if@DivAsExp
6 \DeclareOption{DivAsExponent}{\@DivAsExptrue}
7 \DeclareOption{DivInBrackets}{\@DivAsExpfalse}
8 %%
9 \ExecuteOptions{DivInBrackets}

```

The “`\ProcessOptions*`” command was used here instead of “`\ProcessOptions*`” in order to process the options in the “`\usepackage`” order, rather than in the declaration order

```

10 %%
11 \ProcessOptions*

```

`\DefaultSet` The command `\DefaultSet` makes `\@DefaultSet` to print the given argument enclosed in brackets. It also let `\@DefaultSet@Div` to be the same, but with the word “div” inside the brackets, before the argument.

```

12 %%
13 \newcommand{\DefaultSet}[1]{\def\@DefaultSet{(#1)}}%
14 \def\@DefaultSet@Div{(\mathrm{div};#1)}

```

`\NoDefaultSet` The command `\NoDefaultSet` deletes the content of the two macros `\@DefaultSet` and `\@DefaultSet@Div`, so that the first prints nothing, whilst the second only prints “(div)”.

```

15 \newcommand{\NoDefaultSet}{\let\@DefaultSet=\relax%
16 \def\@DefaultSet@Div{(\mathrm{div})}}

```

Set the default value for `\DefaultSet` to be Ω .

```

17 %%
18 \DefaultSet{\Omega}

```

`\DoNothing@zero` This command does nothing, but strips the character “0” (which *must* immediately follow it, otherwise an error occur) from the input, since it is defined with a “0” attached to its name.

```

19 %%
20 \def\DoNothing@zero0{\relax}

```

`\H` The commands “`\@HSobolevj` and “`\HSobolev@quadra`” prints an “H” in math-mode (via the `\ensuremath` command) with the argument as

a superscript, and, respectively, without subscripts and with the second argument as a subscript. Then they call `\@DefaultSet` in order to print the name of a set inside brackets, or nothing depending on which of the commands `\DefaultSet` and `\NoDefaultSet` has previously been invoked.

```
21 %%
22 \newcommand{\@HSobolev}[1]{\ensuremath{H^{#1}\@DefaultSet}}
23 \def\HSobolev@quadra#1[#2]{\ensuremath{H^{#1}_{#2}\@DefaultSet}}
```

The original command “`\H`” (used to make a type of accent) is saved in `\HAccent`; then `\H` is redefined so that it calls one of the previous two commands, depending on which character follows the command name. If this character is a “0”, then it also invokes the command `\DoNothing@zero`, in order to strip that zero from the input. All this is deferred at the “`\begin{document}`” in order to avoid conflicts with other packages.

```
24 \AtBeginDocument{%
25 \let\HAccent=\H
26 \renewcommand{\H}[1]{%
27 \@ifnextchar0{\HSobolev@quadra{#1}[0]\DoNothing@zero}{%
28 \@ifnextchar[{\HSobolev@quadra{#1}}{\@HSobolev{#1}}}%
29 }}
```

`\Hdiv` The command `\@Hdiv@quadra` is used if the command `\Hdiv` is followed by a square bracket: it simply types “`H`” (in math mode) with a subscript. It also prints “`div`” (in roman) as an exponent and calls `\@DefaultSet` (with the “`DivAsExponent`” option) or calls `\@DefaultSet@Div` (with the “`DivInBraces`” option).

```
30 %%
31 \def\@Hdiv@quadra[#1]{%
32 \if@DivAsExp \ensuremath{H^{\mathrm{div}}_{#1}\@DefaultSet}%
33 \else \ensuremath{H_{#1}\@DefaultSet@Div}\fi%
34 }
```

The “`\Hdiv`” command is defined so that it calls the previous command, or typesets the output itself, similarly to the `\H` command above.

```
35 \newcommand{\Hdiv}{%
36 \@ifnextchar0{\@Hdiv@quadra[0]\DoNothing@zero}{\@ifnextchar[{\@Hdiv@quadra}{%
37 \if@DivAsExp \ensuremath{H^{\mathrm{div}}\@DefaultSet}%
38 \else \ensuremath{H\@DefaultSet@Div}\fi}}%
39 }
```

`\L` First the old command `\L` (which prints an “L” which a bar superimposed) is saved in `\Lbar`. Then `\L` is redefined to print an “L” in math-mode with its argument as a superscript, followed by the output of `\@DefaultSet`. All this is deferred at the “`\begin{document}`” in order to avoid conflicts with other packages.

```
40 %%
41 \AtBeginDocument{%
42 \let\Lbar=\L
43 \renewcommand{\L}[1]{\ensuremath{L^{\#1}\@DefaultSet}}%
44 }
```

`\W` The command `\W` is defined to print a “W” in math-mode with its two arguments as a superscript (separated by a comma) and followed by the output of `\@DefaultSet`. It is deferred at the “`\begin{document}`” in order to avoid conflicts with other packages.

```
45 %%
46 \AtBeginDocument{%
47 \newcommand\W[2]{\ensuremath{W^{\#1,\#2}\@DefaultSet}}
48 }
```

`\D` The command `\D` is defined to print a calligraphic “D” in math-mode, followed by the output of `\@DefaultSet`. In order to allow for the proper treatment of the *prime* symbol which can follow the command, the command `\InsiemeD@Primo` is defined: it is automatically invoked by the command `\D` when followed by a prime. It differs only in that a prime is output *before* invoking `\@DefaultSet`. This is deferred at the “`\begin{document}`” in order to avoid conflicts with other packages.

```
49 %%
50 \def\InsiemeD@Primo'\ensuremath{\mathcal{D}}'\@DefaultSet}}
51 \AtBeginDocument{%
52 \newcommand{\D}{\@ifnextchar'\{ \InsiemeD@Primo}{%
53 \ensuremath{\mathcal{D}}\@DefaultSet}}%
54 }}
```

`\Norm` The command `\Norm` prints its argument surrounded by a double-pipe delimiter of adjustable size. If an optional argument is present, it is used as a whole subscript.

```
55 %%
56 \def\@Norma@Exp#1#2^#3{\ensuremath{\left\| \#1 \right\|_{\#2}^{\#3}}}
57 \def\@Norma#1[#2]{%
58 \@ifnextchar^{\@Norma@Exp{\#1}{\#2}}{\ensuremath{\left\| \#1 \right\|_{\#2}}}%

```

```

59 }
60 \newcommand{\Norm}[1]{%
61 \@ifnextchar[{\@Norma{#1}}{\ensuremath{\left|#1\right|}}%
62 }

```

\SemiNorm The command `\SemiNorm` prints its argument surrounded by a single-pipe delimiter of adjustable size. If an optional argument is present, it is used as a whole subscript.

```

63 %%
64 \def\@SemiNorma@Exp#1#2^#3{\ensuremath{\left|#1\right|_{#2}^{#3}}}
65 \def\@SemiNorma#1[#2]{%
66 \@ifnextchar^{\@SemiNorma@Exp{#1}{#2}}{\ensuremath{\left|#1\right|_{#2}}}%
67 }
68 \newcommand{\SemiNorm}[1]{%
69 \@ifnextchar[{\@SemiNorma{#1}}{\ensuremath{\left|#1\right|}}%
70 }

```

\Scalar The command `\Scalar` requires two arguments; a third optional argument (which is used as a whole subscript) may follow inside square brackets. The output consists of the two arguments separated by a comma and enclosed in a pair of adjustable-size brackets, with the optional argument placed as a subscript.

```

71 %%
72 \def\@ProdottoScalare#1#2[#3]{\ensuremath{\left(#1,#2\right)_{#3}}}
73 \newcommand\Scalar[2]{%
74 \@ifnextchar[{\@ProdottoScalare{#1}{#2}}{\ensuremath{\left(#1,#2\right)}}%
75 }

```

\Crochet The command `\Crochet` requires two arguments; a third optional argument (which is used as a whole subscript) may follow inside square brackets. The output consists of the two arguments separated by a comma and enclosed in a pair of adjustable-size angular-parenthesis, with the optional argument placed as a subscript.

```

76 %%
77 \def\Inner@Crochet#1#2[#3]{\ensuremath{\left\langle#1,#2\right\rangle_{#3}}}
78 \newcommand\Crochet[2]{%
79 \@ifnextchar[{\Inner@Crochet{#1}{#2}}{%
80 \ensuremath{\left\langle#1,#2\right\rangle}}%
81 }

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

| | | |
|--|---|--|
| Symbols | E | <code>\let</code> 15, 25, 42 |
| <code>\@DefaultSet</code> 13, 15, 22, 23, 32, 37, 43, 47, 50, 53 | <code>\else</code> 33, 38 | M |
| <code>\@DefaultSet@Div</code> .. 14, 16, 33, 38 | <code>\ensuremath</code> 22, 23, 32, 33, 37, 38, 43, 47, 50, 53, 56, 58, 61, 64, 66, 69, 72, 74, 77, 80 | <code>\mathcal</code> 50, 53 |
| <code>\@DivAsExpfalse</code> . 7 | <code>\ExecuteOptions</code> . 9 | <code>\mathrm</code> 14, 16, 32, 37 |
| <code>\@DivAsExptrue</code> .. 6 | F | N |
| <code>\@HSobolev</code> .. 22, 28 | <code>\fi</code> 33, 38 | <code>\NeedsTeXFormat</code> . 2 |
| <code>\@Hdiv@quadra</code> 31, 36 | <code>\filedate</code> 3 | <code>\newcommand</code> . 13, 15, 22, 35, 47, 52, 60, 68, 73, 78 |
| <code>\@Norma</code> 57, 61 | <code>\filedescr</code> 3 | <code>\newif</code> 5 |
| <code>\@Norma@Exp</code> . 56, 58 | <code>\FileName</code> 3 | <code>\NoDefaultSet</code> .. <u>15</u> |
| <code>\@ProdottoScalare</code> 72, 74 | <code>\fileversion</code> . . . 3 | <code>\Norm</code> <u>55</u> |
| <code>\@SemiNorma</code> . 65, 69 | H | O |
| <code>\@SemiNorma@Exp</code> 64, 66 | <code>\H</code> <u>21</u> | <code>\Omega</code> 18 |
| <code>\@ifnextchar</code> 27, 28, 36, 52, 58, 61, 66, 69, 74, 79 | <code>\HAccent</code> 25 | P |
| <code>\ </code> 56, 58, 61 | <code>\Hdiv</code> <u>30</u> | <code>\ProcessOptions</code> 11 |
| A | <code>\HSobolev@quadra</code> 23, 27, 28 | <code>\ProvidesPackage</code> 3 |
| <code>\AtBeginDocument</code> .. 24, 41, 46, 51 | I | R |
| C | <code>\if@DivAsExp</code> 5, 32, 37 | <code>\rangle</code> 77, 80 |
| <code>\Crochet</code> <u>76</u> | <code>\Inner@Crochet</code> 77, 79 | <code>\relax</code> 15, 20 |
| D | <code>\InsiemeD@Primo</code> 50, 52 | <code>\renewcommand</code> 26, 43 |
| <code>\D</code> <u>49</u> | L | <code>\right</code> 56, 58, 61, 64, 66, 69, 72, 74, 77, 80 |
| <code>\DeclareOption</code> 6, 7 | <code>\L</code> <u>40</u> | S |
| <code>\def</code> 13, 14, 16, 20, 23, 31, 50, 56, 57, 64, 65, 72, 77 | <code>\langle</code> 77, 80 | <code>\Scalar</code> <u>71</u> |
| <code>\DefaultSet</code> . <u>12</u> , 18 | <code>\Lbar</code> 42 | <code>\SemiNorm</code> <u>63</u> |
| <code>\DoNothing@zero</code> <u>19</u> , 27, 36 | <code>\left</code> 56, 58, 61, 64, 66, 69, 72, 74, 77, 80 | <code>\space</code> 3 |
| | | W |
| | | <code>\W</code> <u>45</u> |

Change History

| | | | |
|------|--|--|---|
| v0.1 | | “Hdiv” command | 1 |
| | General: First release (basic commands) | | 1 |
| v0.2 | | v1.2 | |
| | General: Added package options | General: Superscripts for “Norm” and “SemiNorm” | 1 |
| v0.3 | | v2.0 | |
| | General: Fixed a bug in the command “Scalar” | General: Changed “Scalar” syntax; added “Crochet” . | 1 |
| v0.4 | | v2.1 | |
| | General: Fixed a bug in the options | General: Added copyright notice and changed addresses | 1 |
| v1.0 | | v2.2 | |
| | General: Documentation added | General: Usage of the double-quote character (”) avoided | 1 |
| v1.1 | | | |
| | General: Fixed a bug in the | | |