The tensor*† package for LaTeX2e

Philip G. Ratcliffe‡
Dipartimento di Fisica e Matematica
Università degli Studi dell'Insubria—Como

Abstract

This is a complete revision and extension of Mike Piff's original `tensor` package; it defines two commands for typesetting tensors with mixed upper and lower indices in which the correct horizontal spacing must be observed. Various forms of alignment are available and spaces may be replaced by dots or other symbols. Correct preposing of indices is now made possible while backwards compatibility is maintained. A special-purpose command to typeset nuclides is also defined.

1 Introduction

The use of tensors with mixed upper and lower indices, in which the relative horizontal positions and spacing are significant, is common in both physics and mathematics; for example,

$$\Gamma^{\mu}{}_{\nu\rho}, \quad R^{\mu}{}_{\nu}{}^{\rho}{}_{\sigma} \quad \text{or} \quad \epsilon^{\mu\nu\rho}{}_{\sigma}.$$

The commands defined in this package automatically maintain the correct horizontal positioning. Another common need addressed by this package is the preposing of upper and lower indices, as in

$$_{\mathrm{H}}\langle q', t'|\mathcal{U}(t,t')|q,t\rangle_{\mathrm{H}} \quad \text{or} \quad {}^{14}_{6}\mathrm{C}.$$

Note the correct spacing of the pre-index H in the above example. It should also be noted that constant vertical positioning is maintained for lone indices; consider the following (examine carefully the last lower index o):

$$\mathrm{M}^{o}_{o}|_{o}\mathrm{M} \quad \textit{cf.} \quad \mathrm{M}^{o}_{o}|_{o}\mathrm{M},$$

where the former group was typeset using `\indices` but the latter not.

*This file has version number v2.1, and revision date 2004/12/20.
†Based on and extending the original package of the same name by Mike Piff (1996/06/03).
‡E-mail: philip.ratcliffe@uninsubria.it

2 Usage

Two robust math-mode commands, `\tensor` and `\indices`, are defined (the first of which remains backward compatible with Mike Piff's original definition). A new, robust text- and math-mode command, `\nuclide`, is also defined specifically for typesetting nuclides, as in the above example.

2.1 User commands

`\indices` To produce a mathematical expression (typically a tensor) with mixed upper and lower indices, simply enter ⟨*object*⟩`\indices{^`⟨*sup*$_1$⟩`_`⟨*sub*$_1$⟩`^`⟨*sup*$_2$⟩`_`⟨*sub*$_2$⟩`...}`. Thus, in math mode it is sufficient to type, *e.g.*,

`M\indices{^a_b^{cd}_e}` to obtain $M^{a}{}_{b}{}^{cd}{}_{e}$.

`\tensor` This variant has been retained in a backward compatible form and considerably extended; the syntax for the previous expression is `\tensor{M}{^a_b^{cd}_e}` while the resulting output is identical. The extended form of `\tensor` now defined has an optional argument for indices to be placed *before* the tensor, thus:

`\tensor[^a_b^c_d]{M}{^a_b^c_d}` produces ${}^{a}_{b}{}^{c}_{d}M^{a}{}_{b}{}^{c}{}_{d}$

A fairly robust (if somewhat crude) attempt is made to ensure the correct spacing and skew of the preposed indices with respect to the tensor object itself.

`\indices*` `\tensor*` The two commands have starred forms, which collapse the spacing (*i.e.*, return to standard form). While `\indices*` is clearly redundant (and is included merely for symmetry), `\tensor*` also *right* justifies the *pre*-index strings, so that, *e.g.*, nuclides may be typeset as follows (though see below for a purpose-built command):

`\tensor*[^{14}_6]{\mathrm{C}}{}` produces ${}^{14}_{6}\mathrm{C}$.

For those familiar with the `amsmath` package, this is more-or-less a generalisation of (though *not intended* as a substitute for) the `\sideset` command (which itself is *only valid* for objects defined with `\mathop`). Note that to use `\tensor*` as a substitute for `\sideset`, it is necessary to insert a `\nolimits` command, thus:

`\tensor*[^*_*]{\prod\nolimits}{^*_*}` produces ${}^{*}_{*}\prod^{*}_{*}$.

The output appears identical to that of `\sideset{_*^*}{_*^*}{\prod}`.

`* argument` The `\indices*` and `\tensor*` forms *only*, allow a `*` to also be placed as the first entry in either index-list argument, causing alignment (*left* justification) of the successive pairs of upper and lower indices. A warning is issued if a `*` appears in an argument string of either *non*-starred commands. Thus,

```
\tensor*{M}{*^{i_1}_{m_1}^{i_2}_{m_2}^{i_3}_{m_3}^{i_4}_{m_4}}
```

produces $M^{i_1\ i_2\ i_3\ i_4}_{m_1 m_2 m_3 m_4}$ (*cf.* $M^{i_1 i_2 i_3 i_4}_{m_1 m_2 m_3 m_4}$).

Note that *no warning* is issued for improper pairing of successive indices.

`\indexmarker` In analogy with the `tensind` package, the command `\indexmarker` (by default empty) may redefined (using `\renewcommand`) to introduce a visible place marker for the index spaces (though not all `tensind` functionality is reproduced here); a simple possibility is

```
\renewcommand\indexmarker{\cdot},
```

after which,

```
\tensor{M}{^a_b^c_d}
```

produces $M^{a\cdot c}_{\cdot b\cdot d}$ instead of $M^{a\ c}_{b\ d}$

`\nuclide` This command, available in both math and text modes, is defined with the same purpose and result as the `\isotope` command (from the package of the same name). The syntax is

`\nuclide[`⟨*mass no.*⟩`][`⟨*atomic no.*⟩`]{`⟨*symbol*⟩`}`.

Thus, the earlier example of ${}^{14}_{6}\mathrm{C}$ is obtained with `\nuclide[14][6]{C}` while `\nuclide[4][2]{\alpha}` gives ${}^{4}_{2}\alpha$. As indicated by the square brackets, the ⟨*mass no.*⟩ and ⟨*atomic no.*⟩ arguments are optional. Note that there is a little more space (`1mu`) between the numbers and the chemical symbol than appears in the example constructed manually with `\tensor*`.

All the above-defined commands may be used recursively, *i.e.*, a `\tensor` may occur as an index to another `\tensor` and should behave according to the current superscript–subscript level. Where useful, user commands are defined as 'robust'; thus, they may appear as arguments to `\caption`, `\section` *etc.*

`\nuclideFont` By default, the font for `\nuclide` is `\mathrm`, but `\nuclideFont` may be reset with `\renewcommand` to `\mathsf`, `\mathbf`, `\mathtt` *etc.*, or simply `\relax` (this last has the chemical symbol font default to `\mathit` for correct spacing).

2.2 Caveats

Grouping of multi-token indices should be performed as normal (*i.e.*, via enclosure within a brace pair `{}`). Moreover, owing to the method by which index strings are parsed, any index constructs such as `\mathrm{H}` must also be entirely enclosed in braces, thus: `\indices{_{\mathrm{H}}^x}`.

Spacing is not guaranteed to always appear absolutely perfect, especially between *pre*-pended indices and the tensor object itself. Please also recall, however, that screen viewing often distorts small spaces owing to resolution effects.

2.3 External package requirements

No external packages are required.

2.4 Package conflicts

There are no known conflicts with any standard LaTeX2e packages (a problem with the `color` package in the first version has now been corrected). However, there is obvious incompatibility with the similar `tensind` package.

3 Implementation

3.1 User commands

The `tensor` package now defines three basic user commands:

`\tensor` The first takes three possible arguments (an optional index string to be *preposed*, the tensor object, the index string) and also has a starred form, which suppresses spacing (it is backward compatible with Mike Piff's original version).

```
\DeclareRobustCommand\tensor{%
  \tnsr@Prp
  \@ifstar{\tnsr@Spcfalse\tnsr@Aux}{\tnsr@Spctrue\tnsr@Aux}%
}
```

`\indices` The second command is a "lightweight" form, which is placed immediately *following* the tensor object, takes just one argument (the index string) and also has a starred form (this command was *not* present in the original package).

```
\DeclareRobustCommand\indices{%
  \tnsr@Prp
  \@ifstar{\tnsr@Spcfalse\ndcs@Aux}{\tnsr@Spctrue\ndcs@Aux}%
}
```

`\nuclide` This additional new command takes one direct argument (an optional mass number) and two indirect arguments (an optional atomic number, the chemical symbol—these last two are handled by auxiliary macros). Since usage is common in text, math mode is ensured.

```
\DeclareRobustCommand\nuclide[1][]{%
  \ncld@Mno{#1}%
  \ncld@Aux
}
```

`\nuclideFont` This sets the font for `\nuclide` (the default is `\mathrm`); it may be redefined as, *e.g.*, `\mathsf`, `\mathbf`, `\mathtt`, `\mathit` *etc.*, or even simply `\relax` (this last has the chemical symbol font default to `\mathit` for correct spacing).

```
\newcommand\nuclideFont{\mathrm}
```

3.2 Internal token registers

`\tnsr@Sps` `\tnsr@Sbs` `\tnsr@Spe` `\tnsr@Sbe` The token registers that hold the upper and lower index strings, and the most recent upper and lower index elements respectively:

```
\newtoks\tnsr@Sps
\newtoks\tnsr@Sbs
\newtoks\tnsr@Spe
\newtoks\tnsr@Sbe
```

`\ncld@Mno` This token register temporarily holds the mass number for `\nuclide`.

```
\newtoks\ncld@Mno
```

3.3 Internal switches

`\iftnsr@Spc` The switch to select or suppress index element spacing.

```
\newif\iftnsr@Spc
```

3.4 Internal macros

`\tnsr@Prp` Here we simply reset token registers and the warning macro before commencing.

```
\newcommand\tnsr@Prp{%
  \tnsr@Sps{}%
  \tnsr@Sbs{}%
  \let\tnsr@Wrn\relax
}
```

`\ndcs@Aux` This (lightweight) auxiliary macro for `\indices` takes one argument (an index string); it calls `\tnsr@Set`, prints the indices and then issues any warnings.

```
\newcommand\ndcs@Aux[1]{%
  \def\tnsr@Obj{}
  \tnsr@Set{#1}%
  \tnsr@Fin
  \tnsr@Wrn
}
```

`\tnsr@Aux` This auxiliary macro for `\tensor` takes three possible arguments (an optional pre-index string, the tensor object, the post-index string) and passes everything via `\mathpalette` to `\tnsr@Plt`.

```
\newcommand\tnsr@Aux[3][]{%
  \mathpalette{\tnsr@Plt{#1}{#3}}{#2}%
  \tnsr@Wrn
}
```

`\tnsr@Plt` This takes four arguments (the pre-index string—may be empty, the post-index, the current math style, the tensor object) and calls `\tnsr@Set` separately for both pre- and post-index strings.

```
\newcommand\tnsr@Plt[4]{%
  \def\tnsr@Obj{#3#4}
  \def\reserved@a{#1}%
  \ifx\reserved@a\@empty\else
    \tnsr@Set{#1}%
    \hphantom{{}\tnsr@Fin}%
    \tnsr@Sps\expandafter{%
      \expandafter\tnsr@Krn\expandafter{\the\tnsr@Sps}%
    }%
    \tnsr@Sbs\expandafter{%
      \expandafter\tnsr@Krn\expandafter{\the\tnsr@Sbs}%
    }%
  \fi
  \tnsr@Set{#2}%
```

```
  #4\tnsr@Fin
}
```

`\tnsr@Set` This takes one argument (a pre- or post-index string) and starts processing.

```
\newcommand\tnsr@Set[1]{%
  \let\tnsr@Swx\relax
  \tnsr@Pro#1\tnsr@Err
}
```

`\tnsr@Krn` This has one argument (a processed index string) and inserts the necessary offsets.

```
\newcommand\tnsr@Krn[1]{%
  \settowidth\@tempdima{$\m@th\tnsr@Obj^{#1}\mkern-1mu$}%
  \kern-\@tempdima
  #1
  \settowidth\@tempdima{$\m@th\tnsr@Obj$}%
  \kern\@tempdima
}
```

`\tnsr@Pro` This is the index-string processing macro, it takes one argument (an index string):

```
\newcommand\tnsr@Pro[1]{%
  \ifx#1\tnsr@Err
    \let\tnsr@Nxt\relax
  \else
    \ifx#1*
      \iftnsr@Spc
        \gdef\tnsr@Wrn{%
          \PackageWarning{tensor}{%
            `*' not allowed in argument here, I am ignoring it,%
            \MessageBreak remove it or use `\string\tensor*',%
          }%
        }%
      \else
        \let\tnsr@Swx\tnsr@Swa
      \fi
      \let\tnsr@Nxt\tnsr@Pro
    \else
      \ifx#1^
        \def\tnsr@Nxt{\tnsr@Add{\tnsr@Sps}{\tnsr@Sbs}{\tnsr@Spe}}%
      \else
        \ifx#1_
          \def\tnsr@Nxt{\tnsr@Add{\tnsr@Sbs}{\tnsr@Sps}{\tnsr@Sbe}}%
        \else
          \tnsr@Err
          \let\tnsr@Nxt\tnsr@Pro
        \fi
      \fi
    \fi
  \fi
```

```
  \tnsr@Nxt
}
```

`\tnsr@Swa` Here we flip the state of `\tnsr@Swx` to `\tnsr@Swb`.

```
\newcommand\tnsr@Swa{\let\tnsr@Swx\tnsr@Swb}
```

`\tnsr@Swb` Here we flip the state of `\tnsr@Swx` to `\tnsr@Swa` then calculate and insert the necessary padding for index alignment.

```
\newcommand\tnsr@Swb{%
  \let\tnsr@Swx\tnsr@Swa
  \settowidth\@tempdima{$\m@th\tnsr@Obj{}^{\the\tnsr@Spe}$}%
  \settowidth\@tempdimb{$\m@th\tnsr@Obj{}_{\the\tnsr@Sbe}$}%
  \addtolength\@tempdima{-\@tempdimb}%
  \ifdim\@tempdima=\z@\else
    \ifdim\@tempdima>\z@
      \tnsr@Sbs\expandafter\expandafter\expandafter{%
        \expandafter\the\expandafter\tnsr@Sbs
        \expandafter\kern\the\@tempdima
      }
    \else
      \@tempdima=-\@tempdima
      \tnsr@Sps\expandafter\expandafter\expandafter{%
        \expandafter\the\expandafter\tnsr@Sps
        \expandafter\kern\the\@tempdima
      }
    \fi
  \fi
}
```

`\tnsr@Add` This macro takes four arguments (the token-register target for the next index token, the token-register target for the phantom element, the token-register target for the most-recent element, the next index token). It adds the next index token to the upper or lower string and (if spacing is *on*) a place-holder (`\tnsr@Hph`) of the same size to the lower or upper string, respectively. It also calls `\tnsr@Swx` to flip state (if activated). The use of `\leavevmode` is necessary to avoid conflict with the `color` package.

```
\newcommand\tnsr@Add[4]{%
  #1\expandafter{\the#1\leavevmode{#4}}%
  \iftnsr@Spc
    #2\expandafter{\the#2\tnsr@Hph{#4}}%
  \fi
  #3{\leavevmode{#4}}%
  \tnsr@Swx
  \tnsr@Pro
}
```

`\tnsr@Hph` The place-holder macro, uses `\mathpalette` to call the contents `\tnsr@Mph`:

```
\newcommand\tnsr@Hph{\expandafter\mathpalette\expandafter\tnsr@Mph}
```

`\tnsr@Mph` The place-holder macro contents:

```
\newcommand\tnsr@Mph[2]{%
  \settowidth\@tempdima{$\m@th#1{#2}$}%
  \makebox[\@tempdima][c]{$\m@th#1\indexmarker$}%
}
```

`\indexmarker` The default (blank) placeholder for index spacing:

```
\newcommand\indexmarker{}
```

`\tnsr@Fin` Finally, we put the index strings into place:

```
\newcommand\tnsr@Fin{^{\the\tnsr@Sps}_{\the\tnsr@Sbs}}
```

`\ncld@Aux` This auxiliary macro takes two arguments (an optional atomic number, a chemical symbol). The mass number is passed via `\ncld@Mno`. Math mode is ensured since usage is common in text. The spacing is increased by `1mu` for better appearance.

```
\newcommand\ncld@Aux[2][]{%
  \ensuremath{%
    \tensor*[^{\nuclideFont{\the\ncld@Mno}}_{\nuclideFont{#1}}]%
      {\mkern1mu{\mathit{\nuclideFont{#2}}{}}}{}%
  }%
}
```

`\tnsr@Err` This is invoked in the only error situation considered.

```
\newcommand\tnsr@Err{%
  \PackageError{tensor}{%
    Sub/Superscript items out of order\on@line,
    \MessageBreak some index tokens may now have been lost%
  }{An index string has an extra or missing `^' or `_' token.}%
}
```

Change History

v1.0

General: original version 1

v2.0

General: extended `\tensor`, added `\indices` and `\nuclide`, substituted `\DeclareRobustCommand` for `\newcommand` in user commands, documented and packaged 1

v2.1

`\indexmarker`: added capability to insert place holders 8

`\indices`: added starred form, for symmetry with `\tensor*` 4

`\tnsr@Add`: added `\leavevmode`, to avoid color package conflict . . . 7

`\tnsr@Krn`: altered spacing slightly 6

`\tnsr@Mph`: substituted `\hbox` with `\makebox` 8

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

* argument *2*
\@empty 38
\@ifstar 3, 7
\@tempdima 56, 57, 59, 60, 96, 98–100, 103, 106, 109, 125, 126
\@tempdimb 97, 98

A

\addtolength 98

D

\DeclareRobustCommand 1, 5, 9
\def . . . 26, 36, 37, 80, 83

E

\else 38, 65, 74, 78, 81, 84, 99, 105
\ensuremath 131
\expandafter . . . 41, 42, 44, 45, 101–103, 107–109, 115, 117, 123

F

\fi 47, 76, 87–90, 111, 112, 118

G

\gdef 68

H

\hphantom 40

I

\ifdim 99, 100
\iftnsr@Spc . 19, 67, 116
\ifx . . . 38, 63, 66, 79, 82
\indexmarker *3*, 126, 128
\indices *2*, 5
\indices* *2*

K

\kern . . . 57, 60, 103, 109

L

\leavevmode . . . 115, 119
\let 23, 52, 64, 75, 77, 86, 93, 95

M

\m@th 56, 59, 96, 97, 125, 126
\makebox 126
\mathit 133
\mathpalette . . . 32, 123
\mathrm 13
\MessageBreak . . 71, 139
\mkern 56, 133

N

\ncld@Aux 11, 130
\ncld@Mno . . 10, 18, 132
\ndcs@Aux 7, 25
\newcommand 13, 20, 25, 31, 35, 51, 55, 62, 93, 94, 114, 123, 124, 128–130, 136
\newif 19
\newtoks 14–18
\nuclide *3*, 9
\nuclideFont *3*, 13, 132, 133

O

\on@line 138

P

\PackageError 137
\PackageWarning . . . 69

R

\relax 23, 52, 64
\reserved@a 37, 38

S

\settowidth 56, 59, 96, 97, 125
\string 71

T

\tensor . . . 1, *2*, 71, 132
\tensor* *2*
\the 42, 45, 96, 97, 102, 103, 108, 109, 115, 117, 129, 132
\tnsr@Add . . 80, 83, 114
\tnsr@Aux 3, 31
\tnsr@Err 53, 63, 85, 136
\tnsr@Fin 28, 40, 49, 129
\tnsr@Hph 117, 123
\tnsr@Krn . . . 42, 45, 55
\tnsr@Mph 123, 124
\tnsr@Nxt 64, 77, 80, 83, 86, 91
\tnsr@Obj 26, 36, 56, 59, 96, 97
\tnsr@Plt 32, 35
\tnsr@Pro . . 53, 62, 121
\tnsr@Prp 2, 6, 20
\tnsr@Sbe . . . 14, 83, 97
\tnsr@Sbs 14, 22, 44, 45, 80, 83, 101, 102, 129
\tnsr@Set 27, 39, 48, 51
\tnsr@Spcfalse . . . 3, 7
\tnsr@Spctrue 3, 7
\tnsr@Spe . . . 14, 80, 96
\tnsr@Sps 14, 21, 41, 42, 80, 83, 107, 108, 129
\tnsr@Swa . . . 75, 93, 95
\tnsr@Swb 93, 94
\tnsr@Swx 52, 75, 93, 95, 120
\tnsr@Wrn 23, 29, 33, 68

Z

\z@ 99, 100