deleq – a LaTeX Macro for Partial Numbering of Equations*

Mats Dahlgren†

1997/06/05

Abstract

To enable a more flexible equation numbering, especially "partial" equation numbers ('3a', '3b' *etc.*), the deleq package has been developed. It can produce partial equation numbers intermixed with ordinary equation numbers also in an `eqnarray`-like environment, the intermixing can occur within one environment. References to a partially numbered equation can be both the complete equation number ('3b') or only the main equation number ('3'). Furthermore, equation numbers can be recycled without disturbing the ordinary equation numbers. The package also provides commands for putting commentatory text in an `eqnarray` environment. Both standard LaTeX class options `leqno` and `fleqn`[1] work with deleq.

1 Introduction

deleq is a LaTeX package which makes partial numbering of equations possible. It is meant to be used when numbering such as 3a, 3b, *etc.* is desired. The default is to give an equation number like '3a' without period and with the 'a' typeset in roman font. It can be used in `article` as well as `book` and `report` document classes. The examples given below are valid for the `article` class. deleq is fully compatible with the `leqno` documentclass option and almost fully with the `fleqn` class option. Equations can be numbered either 3, 3a, 3b, ..., or 2, 3a, 3b, ... Also, equation numbers within `eqnarray`-like environments can be numbered 3, 3a, 3b, ... or starting at 3a, which can follow both after equation 2b or 3. An `eqnarray`-like environment can have its first equation numbered 3b if the nearest previous equation is 3a. Within one `eqnarray`-like environment, equations can be numbered 3a, 3b, 4, 4a, 4b, 5a, 5b, *etc.*, and also not numbered lines are possible ('`\nonumber`' works with some limitations). Furthermore, equation numbers can be "recycled". If equation 3 is repeated after equation 8, it can still have the numbering set to 3, and be followed by (a new) equation 9. "Recycled" equation numbers can receive partial numbers (3a, 3b, ...); partially numbered equations can also be "recycled" (but at this stage, the latter can not receive new partial numbers neatly, it will come out like '3ba' if eqution 3b is the "recycled" equation). Two commands are supplied which enable the user to write commentary texts in `eqnarray`-like environments without interfering with the alignment.

This userguide is also available in `.pdf`-format on the internet. It is found from my LaTeX web page: `http://www.homenet.se/matsd/latex/`

*This document describes deleq version 4.41a, last revised 1997/06/05

†Email: `matsd@sssk.se` Web: `http://www.homenet.se/matsd/`

[1] Not all commands of deleq work with the `fleqn` option.

1.1 History

`deleq.sty` was originally written for LaTeX 2.09, and contained several commands which became obsolete with the introduction of LaTeX 2_ε. These commands have been removed. The first release for LaTeX 2_ε was v.4.0, of Oct. 14, 1994. The December 1994 release (v.4.1, Dec. 27, 1994) was the first which was compatible with the `docstrip` utility of Johannes Braams, Denys Duchier and Frank Mittelbach. In release v.4.2 new abilities to reference entire `deqarr` and `ddeqar` environments and enhanced abilities for recycled equation numbers were introduced. Also in deleq v.4.2 this documentation was revised and some internal commands compacted. A previous release, v.4.3, was made nessecary because the changed handeling of arguments in connection with the options in LaTeX 2_ε of 1995/06/01. In release v.4.4, compatibility with the LaTeX class option `fleqn` was introduced.[2] In v.4.41 only some changes to the documentation have been introduced.

deleq version 4.41 has been tested with LaTeX 2_ε of 1997/06/01 running TeX 3.14159 in MiKTeX 1.07 under Windows 95. Please send bug reports (see below), corrections, additions, suggestions, *etc.* to me at `matsd@sssk.se`. Version 3.02 is the last (non-supported) version for LaTeX 2.09. (Command names are a mixture of LaTeX, Swedish and a bit of the author's fantasy...)

1.2 Known Problems

`\eqreqno` `\deqreqno` `\ddeqreqno` `deqarr` `ddeqar` `\jotbaseline`

- When using `\eqreqno`, `\deqreqno`, and `\ddeqreqno` on the last line in a `deqarr`, `ddeqar`, `eqnarray`, or `eqnarray*`, there will be an extra blank line with an equation number (not in the `*` form) at the end of the environment. To avoid this problem, use
  `\eqreqno[-\jotbaseline]{FOO} \nonumber`
  instead of `\eqreqno{FOO}` (the analogues for `\deqreqno` and `\ddeqreqno` are obvious).

2 Userguide

2.1 Requirements

The file `deleq.sty` must be available in the user's `TEXINPUTS` directories. It requires LaTeX 2_ε of 1996/12/01 (or newer).

2.2 Usage

The package is included by stating
`\usepackage{deleq}`
In the document preamble. The documentclass option `leqno` is fully supported, and the option `fleqn` is recognized by the environments defined in deleq. Other class or package options do not have any effect on deleq.

2.3 Environments

The package defines the following five environments:

`deqn` `deqn` is an equation environment for automatic numbering of the first equation of type '3a';

`ddeqn` `ddeqn` is an equation used for automatic numbering of the equations following the one in `deqn`, see also details below;

`deqarr` `deqarr` is an environment for automatic partial equation numbering in `eqnarray`-like environment

[2] `fleqn` incompatibility was pointed out by Peter Kruijt (`peterk@wfw.wtb.tue.nl`)

with `deqn` functionality for the first equation;

`ddeqar` `ddeqar` similar to `deqarr` environment, but with `ddeqn` behaviour for the first equation, see details below;

`deqrarr` `deqrarr` this is an '`eqnarray`'-like environment specially designed for recycled equation numbers.

2.4 Commands

The package defines the following fourteen user commands:

```
\nydeqno
\heqno
\reqno{FOO}
\rndeqno{FOO}
\rdeqno{FOO}
\eqreqno[len]{FOO}
\deqreqno[len]{FOO}
\ddeqreqno[len]{FOO}
\arrlabel{FOO}
\where
\rem{text}
\nydeleqno
\deleqno
\jotbaseline
```

2.5 Notice

`\nydeqno` `\heqno` `\arrlabel` `deqarr` `ddeqarr`

1. The use of `\arrlabel` may give unexpected results if any of the commands `\nydeqno` or `\heqno` is used in the same `deqarr` or `ddeqar` environment.
2. If you use `\nydeqno`, `\heqno`, `\reqno`, `\rndeqno`, `\rdeqno`, `\deqreqno`, `\ddeqreqno`, `\nydeleqno`, or `\deleqno` in combination with `\label{FOO}`, `\label{FOO}` should appear after the deleq-command;
3. If you change the appearance of equation numbers (*e.g.* use '[2]' instead of '(2)'), deleq's commands may not typeset equations with partial numbers like your ordinary equations.
4. The class option `fleqn` is not recognized by the commands `\reqno`, `\rndeqno`, `\rdeqno`, `\nydeleqno`, and `\deleqno`.

3 Syntax

Here follows a more detailed description of the different environments and commands.

`deqn` `\begin{deqn} ... \end{deqn}`

Typesets an equation just like `\begin{equation} ... \end{equation}` does, but gives it a number such as '3a' instead of '3' (always 'a'). It steps the main equation counter.

`ddeqn` `\begin{ddeqn} ... \end{ddeqn}`

Typesets an equation just like `\begin{equation} ... \end{equation}` does, but gives it a number such as '3b', '3c', ... instead of '3', '4', ... when following after another partially

numbered equation. When following after an ordinarily numbered equation ('3') it gives a partial equation number with the last used ordinary equation number, *e.g.* '3a'. Can be made to produce the result of `deqn` environment by the use of `\nydeqno` (see below). `ddeqn` does not step the main equation counter.

`deqarr` `\begin{deqarr} ... \end{deqarr}`

Typesets an equation array just like `\begin{eqnarray} ... \end{eqnarray}` does, but gives the first numbered equation a number like '3a' instead of '3' (always 'a'), and the following '3b', '3c' *etc.* It steps the main equation counter. The `\nonumber` command works just like in the `eqnarray` environment on the partial equation counter. Thus, if you use `\nonumber` on all lines in an `deqarr` environment (or on all lines before or between `\heqno` or `\nydeleqno`) the equation numbering in your document will be incorrect. There is no warning issued by LaTeX if this happends. If you want blank lines and then a `\heqno`'ed equation followed by partially numbered equations, use the `ddeqar` environment instead. The same is true for `\nydeqno`'ed equations.

`ddeqar` `\begin{ddeqar} ... \end{ddeqar}`

Typesets an equation array just like `\begin{eqnarray} ... \end{eqnarray}` does, but gives the equation numbers such as '3a' if following after an equation numbered '3' (ordinary equation number) and numbers such as '3c' if following after a partially numbered equation '3b'. Can be made to produce the result of `deqarr` environment by the use of `\nydeqno`. `ddeqar` does not step the main equation counter. The `\nonumber` command works like in the `deqarr` environment.

`deqrarr` `\begin{deqrarr} ... \end{deqrarr}`

This environment is meant to host recycled equations and has `eqnarray` structure. By default, it issues no equation number at all on a line which ends with `\\`. The use of `\label{FOO}` within `deqrarr` returns the present partial equation number of recycled equations (just like with `\rndeqno` and `\rdeqno`). Note that neither `\heqno` nor `\nydeleqno` gives any equation numbers at all in the `deqrarr` environment.

`\nydeqno` `\nydeqno`

Used within `deqarr` and `ddeqar` environments to step the main equation number by one and reset the partial equation number to 'a'; thus, `\nydeqno` gives equation number '4a' when following after equation '3c'.

`\heqno` `\heqno`

Used within `deqarr` and `ddeqar` environments to step the main equation number by one and to produce an ordinary equation number; thus, gives equation number '4' when following equation '3c'. Equations following the `\heqno`-ed will be partially numbered with the `\heqno`-ed equation's number as the main number, *e.g.* '4a' (unless it has a `\nydeqno` command, which in this case would produce the equation number '5a').

`\reqno` `\reqno{FOO}`

Is used when repeating an equation with its original number, "recycling" the eqaution number. `\reqno` takes the argument `FOO`, which has to be defined by a `\label{FOO}` in the original equation. It can only be used within `$$ ... $$`. It does not affect the equation number counter, nor the ordinary partial equation number counter. However, it resets the partial equation number counter for the `\rndeqno`, `\rdeqno`, `\deqreqno`, and `\ddeqreqno` commands. If used with a `\label{FOOO}` command, the `.aux`-file will only contain the page

number of label FOOO. This command is not compatible with the class option fleqn. Instead of $$ ... \reqno{FOO} $$, use a one-line deqrarr environment with the commands \eqreqno[-\jotbaseline]{FOO} \nonumber at the end.

\rndeqno `\rndeqno{FOO}`

Adds a partial equation number to an old equation number, specified by the FOO label. If FOO refers to equation '4', \rdeqno{FOO} will result in equation number '4a' (always 'a'). It can only be used within $$ ... $$. It does not affect the equation number counter, nor the ordinary partial equation number counter. However, it resets the partial equation number counter for the \rndeqno, \rdeqno, \deqreqno, and \ddeqreqno commands. The argument of \rndeqno follows the same rules as that of \reqno. When used with the \label{FOOO} command, a reference to FOOO will only return the partial equation number (and the page number). To make a complete reference to an equation which has a \rndeqno command, say \ref{FOO}\ref{FOOO}. This command is not compatible with the class option fleqn. Instead of $$ ... \rndeqno{FOO} $$, use a one-line deqrarr environment with the commands \deqreqno[-\jotbaseline]{FOO} \nonumber at the end.

\rdeqno `\rdeqno{FOO}`

Same as \rndeqno{FOO} but without resetting any equation number counter and giving consecutive partial equation numbers ('4b', '4c', *etc.*). **Note:** There is nothing preventing the repeated use of \rndeqno{FOO} and \rdeqno{FOO} for the same label FOO. This will result in numbering such as '3a' (if FOO refers to equation '3') occuring several times. This command is not compatible with the class option fleqn. Instead of $$ ... \rdeqno{FOO} $$, use a one-line deqrarr environment with the commands \ddeqreqno[-\jotbaseline]{FOO} \nonumber at the end.

\eqreqno `\eqreqno[`*len*`]{FOO}`

This command is the array-like version of \reqno{FOO} and is used in much the same way. The optional argument *len* is a length which is added between consecutive rows in the array-like structure. When using \eqreqno no '\\' should be issued at the end of the line, it is embedded in the command. This is giving a strange appearance if \eqreqno is used on the last line of the array-like sturcture, namely an extra blank line (with equation number). To avoid this problem, specify a negative length for *len*, preferably '-\jotbaseline' (se below), and issue a '\nonumber' afterwards.

\deqreqno `\deqreqno[`*len*`]{FOO}`

This is the '\rndeqno' version of '\eqreqno' and is used as the former with the latter's abilities at the end of a line.

\ddeqreqno `\ddeqreqno[`*len*`]{FOO}`

This is the '\rdeqno' version of '\eqreqno' and is used as the former with the latter's abilities at the end of a line.

\arrlabel `\arrlabel{FOO}`

This command is a version of LaTeX's ordinary \label{FOO} command meant to be used in deqarr and ddeqar environments. In these environments, a \label{FOO} command gives a reference to the specific equation, *e.g.* 3b, whereas \arrlabel{FOO} in the same position gives a reference containing only the main equation number, 3 in this example. For restrictions, see the section "Notice" above. The four commands \eqreqno, \deqreqno, \ddeqreqno, and \arrlabel were created after inspiration from Larry Jones (schnuff@mit.edu).

\where `\where`

The command `\where` typsets the text "where" (in the default LR-font) flush left on a seperate row in `eqnarray`, `deqarr`, `ddeqar`, and `deqrarr` environments, and preserves the environment's alignment for rows to come.

`\remtext` `\rem{`*text*`}`

The command `\rem{`*text*`}` typsets the text "*text*" (in the default LR-font) flush left on a seperate row in `eqnarray`, `deqarr`, `ddeqar`, and `deqrarr` environments, and preserves the environment's alignment for rows to come. `\where` is equivalent to `\rem{where}`.

`\nydeleqno` `\nydeleqno`

Gives a new partial equation number when used within `$$ ... $$`. Thus, writing `$$ ... \nydeleqno $$` is equivalent to writing `\begin{deqn} ... \end{deqn}`. This command is not compatible with the class option `fleqn`, use `\begin{deqn} ... \end{deqn}` instead.

`\deleqno` `\deleqno`

Gives a partial equation number when used within `$$ ... $$`. Thus, writing `$$ ... \deleqno $$` is equivalent to writing `\begin{ddeqn} ... \end{ddeqn}`. This command is not compatible with the class option `fleqn`, use `\begin{ddeqn} ... \end{ddeqn}` instead.

`\jotbaseline` `\jotbaseline`

This is a rubber length which is set to be the sum of `\baselinsekip` and `\jot` each time a `deqarr`, `ddeqar`, or a `deqrarr` environment is enterd. It is meant to be used with the `\eqreqno` command and its relatives to eliminate the problems when these commands are used on the last line of the environment. (The length `\baselineskip` + `\jot` is the length LaTeX skips between two empty lines in an '`eqnarray`' environment and its derivates defined in deleq.)

4 Example

After running LaTeX on `deleq.ins`, there is an example avaliable in the file `delex.tex`, which makes use of all the environments and commands defined in the deleq package. Typset it and see with your own eyes what the results are!

5 Sending a Bug Report

deleq is most likely to contain bugs. Reports of bugs in the package are most welcome. Before filing a bug report, please take the following actions:

1. Ensure your problem is not due to your inputfile;
2. Ensure your problem is not due to your own package(s) or class(es);
3. Ensure your problem is not covered in the section "Known Problems" above;
4. Try to locate the problem by writing a minimal LaTeX input file which reproduces the problem. Include the command
   `\setcounter{errorcontextlines}{999}`
   in your input;
5. Run your file through LaTeX;
6. Send a description of your problem, the input file and the log file via e-mail to:
   `matsd@sssk.se`

Enjoy your LaTeX!

mats d.

6 The Code

For the interested reader(s), here is a short description of the code.

First, the package is to identify itself.

```
\NeedsTeXFormat{LaTeX2e}[1996/12/01]
\ProvidesPackage{deleq}[1997/07/07 v.4.41 Partial equation numbering]
\def\deleqver{\textsf{deleq} version 4.41 (July~7, 1997)}
```

Then, we need to define the counters used for the partial part of the equation number. The counter `Deleq` is used for "recycled" equation numbers, hence no dependence on any other counter.

```
\newcounter{deleq}[equation]
\newcounter{Deleq}
\newlength{\jotbaseline}
\renewcommand{\thedeleq}{\ensuremath{\theequation\mathrm{\alph{deleq}}}}
\renewcommand{\theDeleq}{\ensuremath{\mathrm{\alph{Deleq}}}}
\def\@deleqnnum{(\thedeleq)}
\def\@deleqrnum#1{(\ref{#1}\theDeleq)}
\def\@eqrnum#1{(\ref{#1})}
```

The parts making up the `\\` command in the multiline environments are similar to those used by the standard environments (at least how they looked some time ago):

```
\def\@deqncr{{\ifnum0=`}\fi\@ifstar{\global\@eqpen\@M
    \@ydeqncr}{\global\@eqpen\interdisplaylinepenalty \@ydeqncr}}
\def\@ydeqncr{\@ifnextchar [{\@xdeqncr}{\@xdeqncr[\z@]}}
\def\@xdeqncr[#1]{\ifnum0=`{\fi}\@@deqncr
   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}}
\def\@@deqncr{\let\@tempa\relax
    \ifcase\@eqcnt \def\@tempa{& & &}\or \def\@tempa{& &}%
      \else \def\@tempa{&}\fi
     \@tempa \if@eqnsw\@deleqnnum\stepcounter{deleq}\fi
     \global\@eqnswtrue\global\@eqcnt\z@\cr}
\newcommand{\@xeqrcr}[2]{\ifnum0=`{\fi}\@@eqrcr{#2}
   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}}
\def\@@eqrcr#1{\let\@tempa\relax
    \ifcase\@eqcnt \def\@tempa{& & &}\or \def\@tempa{& &}%
      \else \def\@tempa{&}\fi
     \@tempa \if@eqnsw\@eqrnum{#1}\fi
     \global\@eqnswtrue\global\@eqcnt\z@\cr}
\newcommand{\@xdeqrcr}[2]{\ifnum0=`{\fi}\@@deqrcr{#2}
   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}}
\def\@@deqrcr#1{\let\@tempa\relax
    \ifcase\@eqcnt \def\@tempa{& & &}\or \def\@tempa{& &}%
      \else \def\@tempa{&}\fi
     \@tempa \if@eqnsw\@deleqrnum{#1}\fi
     \global\@eqnswtrue\global\@eqcnt\z@\cr}
\def\@deqrarcr{{\ifnum0=`}\fi\@ifstar{\global\@eqpen\@M
    \@ydeqrarcr}{\global\@eqpen\interdisplaylinepenalty \@ydeqrarcr}}
\def\@ydeqrarcr{\@ifnextchar [{\@xdeqrarcr}{\@xdeqrarcr[\z@]}}
\def\@xdeqrarcr[#1]{\ifnum0=`{\fi}\@@deqrarcr
   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}}
\def\@@deqrarcr{\let\@tempa\relax
    \ifcase\@eqcnt \def\@tempa{& & &}\or \def\@tempa{& &}%
      \else \def\@tempa{&}\fi
     \@tempa \global\@eqnswtrue\global\@eqcnt\z@\cr}
```

Here the definitions of the user commands are.

```
\newcommand{\nydeleqno}{\stepcounter{equation}\stepcounter{deleq}
  \let\@currentlabel\thedeleq \eqno (\thedeleq)}
\newcommand{\deleqno}{\refstepcounter{deleq} \let\@currentlabel\thedeleq
  \eqno (\thedeleq)}
\newcommand{\reqno}[1]{\setcounter{Deleq}{-1}\refstepcounter{Deleq}
   \eqno (\ref{#1})}
\newcommand{\rdeqno}[1]{\refstepcounter{Deleq}
   \eqno (\ref{#1}\theDeleq)}
\newcommand{\rndeqno}[1]{\setcounter{Deleq}{0}\refstepcounter{Deleq}
   \eqno (\ref{#1}\theDeleq)}
\newcommand{\eqreqno}[2][0pt]{{\ifnum0=`}\fi\@ifstar{\global\@eqpen\@M
    \@xeqrcr{#1}{#2}}{\global\@eqpen\interdisplaylinepenalty
    \@xeqrcr{#1}{#2}}}
\newcommand{\deqreqno}[2][0pt]{\setcounter{Deleq}{0}\refstepcounter{Deleq}
    {\ifnum0=`}\fi\@ifstar{\global\@eqpen\@M
    \@xdeqrcr{#1}{#2}}{\global\@eqpen\interdisplaylinepenalty
    \@xdeqrcr{#1}{#2}}}
\newcommand{\ddeqreqno}[2][0pt]{\refstepcounter{Deleq}
    {\ifnum0=`}\fi\@ifstar{\global\@eqpen\@M
    \@xdeqrcr{#1}{#2}}{\global\@eqpen\interdisplaylinepenalty
    \@xdeqrcr{#1}{#2}}}
\newcommand{\arrlabel}[1]{\let\@currentlabel\theequation \label{#1}}
\newcommand{\nydeqno}{\stepcounter{equation}\stepcounter{deleq}}
\newcommand{\heqno}{\stepcounter{equation}}
\def\where{\let\@tempa\relax \def\@tempa{& & &}
     \@tempa {\hbox to .01\p@{}\rlap{\hskip -\displaywidth where}}\cr}
\def\rem#1{\let\@tempa\relax \def\@tempa{& & &}
     \@tempa {\hbox to .01\p@{}\rlap{\hskip -\displaywidth #1}}\cr}
```

The five environments are slight modifications of the corresponding LaTeX standard environments. The main difference lies in which counter(s) is stepped and which internals are used to finish off the lines. Three of them have large parts in common, parts which are put in one macro '`\@dlqnv`':

```
\def\@dlqnv{\setlength{\jotbaseline}{\baselineskip}%
  \addtolength{\jotbaseline}{\jot}  \global\@eqnswtrue\m@th
  \global\@eqcnt\z@\tabskip\@centering
  $$\halign to\displaywidth\bgroup\@eqnsel\hskip\@centering
  $\displaystyle\tabskip\z@{##}$&\global\@eqcnt\@ne
  \hskip 2\arraycolsep \hfil${##}$\hfil
  &\global\@eqcnt\tw@ \hskip 2\arraycolsep $\displaystyle\tabskip\z@{##}$\hfil
   \tabskip\@centering&\llap{##}\tabskip\z@\cr}
\newenvironment{deqarr}{\stepcounter{equation}\stepcounter{deleq}
  \let\@currentlabel\thedeleq  \let\\\@deqncr  \@dlqnv}
{\@@deqncr\egroup
     \global\advance\c@deleq\m@ne$$\global\@ignoretrue}
\newenvironment{ddeqar}{\stepcounter{deleq}
  \let\@currentlabel\thedeleq  \let\\\@deqncr  \@dlqnv}
{\enddeqarr}
\newenvironment{deqn}{$$\refstepcounter{equation}\stepcounter{deleq}
   \let\@currentlabel\thedeleq}
  {\eqno \hbox{\@deleqnnum} $$\global\@ignoretrue}
\newenvironment{ddeqn}{$$\refstepcounter{deleq} \let\@currentlabel\thedeleq}
  {\enddeqn}
```

```
\newenvironment{deqrarr}{\let\@currentlabel\theDeleq
  \let\\\@deqrarcr   \@dlqnv}
{\@deqrarcr   \egroup  $$\global\@ignoretrue}
```

If the user wants the equation numbers to the left, we have to modify some of the commands and internals defined above. This is done in a `\DeclareOption`-call, but first we set a switch to test for the `leqno` option. Initially, it is set to `F` and then changed to `T` if the `leqno` option is in effect.

```
\newif\ifl@qn \l@qnfalse
\DeclareOption{leqno}{%
\global\l@qntrue%
\renewcommand{\@deleqnnum}{\hbox to .01\p@{}\rlap{\reset@font\rmfamily
  \hskip -\displaywidth(\thedeleq)}}
\renewcommand{\nydeleqno}{\stepcounter{equation}\stepcounter{deleq}
  \let\@currentlabel\thedeleq \leqno (\thedeleq)}
\renewcommand{\deleqno}{\refstepcounter{deleq} \let\@currentlabel\thedeleq
  \leqno (\thedeleq)}
\renewcommand{\reqno}[1]{\setcounter{Deleq}{-1}\refstepcounter{Deleq}
   \leqno (\ref{#1})}
\renewcommand{\rdeqno}[1]{\refstepcounter{Deleq}
   \leqno (\ref{#1}\theDeleq)}
\renewcommand{\rndeqno}[1]{\setcounter{Deleq}{0}\refstepcounter{Deleq}
   \leqno (\ref{#1}\theDeleq)}
\renewcommand{\@eqrnum}[1]{\hbox to .01\p@{}\rlap{\reset@font\rmfamily
        \hskip -\displaywidth(\ref{#1})}}
\renewcommand{\@deleqrnum}[1]{\hbox to .01\p@{}\rlap{\reset@font\rmfamily
        \hskip -\displaywidth(\ref{#1}\theDeleq)}}
 }
```

The `fleqn` option is mainly a 'deleq-ification' of the LaTeX 2_ε file `fleqn.clo` (1995/06/26 v1.3g). If both `leqno` and `fleqn` options are in use, the length `\mathindent` should be increased to allow space for the letters of the partial equation numbers. For the multiline environments, it is enough to make changes to the internal `\@dlqnv`.

```
\DeclareOption{fleqn}{%
\ifl@qn \AtBeginDocument{\addtolength{\mathindent}{1em}} \fi
\renewenvironment{deqn}%
  {\@beginparpenalty\predisplaypenalty
   \@endparpenalty\postdisplaypenalty
   \refstepcounter{equation}\stepcounter{deleq}
   \let\@currentlabel\thedeleq%
   \trivlist \item[]\leavevmode
     \hb@xt@\linewidth\bgroup $\m@th% $
     \displaystyle
     \hskip\mathindent}%
    {$\hfil % $
     \displaywidth\linewidth\hbox{\@deleqnnum}%
     \egroup
   \endtrivlist}
\renewenvironment{ddeqn}%
  {\@beginparpenalty\predisplaypenalty
   \@endparpenalty\postdisplaypenalty
   \refstepcounter{deleq}
   \let\@currentlabel\thedeleq%
   \trivlist \item[]\leavevmode
```

```
    \hb@xt@\linewidth\bgroup $\m@th% $
    \displaystyle
    \hskip\mathindent}%
   {\enddeqn}
\renewcommand{\@dlqnv}{%
  \setlength{\jotbaseline}{\baselineskip}
  \addtolength{\jotbaseline}{\jot}
  \global\@eqcnt\z@
  \tabskip\mathindent
  \setlength\abovedisplayskip{\topsep}%
  \ifvmode
    \addtolength\abovedisplayskip{\partopsep}%
  \fi
  \addtolength\abovedisplayskip{\parskip}%
  \setlength\belowdisplayskip{\abovedisplayskip}%
  \setlength\belowdisplayshortskip{\abovedisplayskip}%
  \setlength\abovedisplayshortskip{\abovedisplayskip}%
  $$\everycr{}\halign to\linewidth% $$
  \bgroup
    \hskip\@centering
    $\displaystyle\tabskip\z@skip{##}$\@eqnsel&%
    \global\@eqcnt\@ne \hskip \tw@\arraycolsep \hfil${##}$\hfil&%
    \global\@eqcnt\tw@ \hskip \tw@\arraycolsep
    $\displaystyle{##}$\hfil \tabskip\@centering&%
    \global\@eqcnt\thr@@
    \hb@xt@\z@\bgroup\hss##\egroup\tabskip\z@skip\cr}
}
```

If other options were asked for, the package should make these options 'unused'. Then, last, the option(s) is (are) processed.

```
\DeclareOption*{\OptionNotUsed}
\ProcessOptions
```

That is all. Happy TEX-ing!