Subnumbering of equations[*]

Donald Arsenau Johannes Braams

2004/04/15

1 Introduction

Sometimes it is necessary to be able to refer to subexpressions of an equation. In order to do that these subexpressions should be numbered. In standard LaTeX there is no provision for this. To solve this problem Stephen Gildea once wrote `subeqn.sty` for LaTeX 2.09; Donald Arsenau rewrote the macros and Johannes Braams made them available for LaTeX 2$_\varepsilon$.

Note that this package is *not* compatible with the package `subeqnarray`, written by Johannes Braams.

This package can be used together with the LaTeX options leqno and fleqn.

2 Available environments

`subeqations` Inside the subeqations environment LaTeX's equation environments such as equation and eqnarray are numbered as subexpressions. At the same time the number of the (main) equation is kept the same.

`subeqnarray` `\begin{subeqnarray}` works like `\begin{subequations}\begin{eqnarray}`, but saves typing. A `\label` command given at the very beginning of the first entry defines a `label` for the overall equation number, as if you had typed `\begin{subequations}\label{xxx}\begin{eqnarray}`.

3 Available commands

`\thesubequation` The command `\thesubequation` controls the labelling of the subexpressions of an equation. You can change the labelling by redefining this command, but the names of the counters may be confusing: The sub-number is given by counter `equation`, while the overall equation number is given by `mainequation`.

There are two ways to reference the overall equation number: through its value, as in `\Roman{mainequation}`, or through `\themainequation`, which gives the text of the normal `\theequation`. Refer to the local sub-number through the value of the `equation` counter, as in `\alph{equation}`. The default numbering is like 13c, given by:

```
\newcommand*{\thesubequation}{\themainequation\alph{equation}}
```

[*] This file has version number v2.0b, last revised 2004/04/15.

Some alternatives:
A number such as 13.C is achieved by

```
\newcommand*{\thesubequation}{\themainequation.\Alph{equation}}
```

A number such as 13-iii is achieved by

```
\newcommand*{\thesubequation}{\themainequation-\roman{equation}}
\newcommand*{\thesubequation}{\themainequation.\Alph{equation}}
```

When the document class which is used has declared

```
\renewcommand{\@eqnnum}{\theequation}
\renewcommand{\theequation}{(\arabic{equation})}
```

which puts parentheses around *all* equation numbers, including those produced by the `\ref` command, you can use:

```
\newcommand*{\thesubequation}{(\arabic{mainequation}\alph{equation})}
```

4 The implementation

```
⟨*package⟩
```

`subeqations` Within the subequations the equation numbers consist of two parts. The first part is a representation of the current value of the `equation` counter when the environment is entered, ie the number of the equation; the second part indicates the number of the subexpression of the equation.

```
\newenvironment{subequations}{%
```

First we update the `equation` counter,

```
  \refstepcounter{equation}%
```

then we save its current value in `\c@mainequation` and define `\themainequation` to be the current representation of the `equation` counter.

```
  \mathchardef\c@mainequation\c@equation
  \protected@edef\themainequation{\theequation}%
```

Then we change the representation of the `equation` counter to represent the subexpression number. Finally we set the `equation` counter to zero as we use it for counting the subexpressions.

```
  \let\theequation\thesubequation
  \global\c@equation\z@
  }{%
```

When the environment is finished we restore the value ot the `equation` counter.

```
  \global\c@equation\c@mainequation
  \global\@ignoretrue
  }
```

`\thesubequation` By default the subexpressions will be numbered with lower case letters. The representation of the `equation` counter also includes the saved value of the `equation` counter. This can be changed by redefining this command.

```
\newcommand{\thesubequation}{\themainequation\alph{equation}}
```

`subeqnarray`

```
\newenvironment{subeqnarray}{%
  \subequations
  \@ifnextchar\label{\@lab@subeqnarray}{\eqnarray}
  }{%
  \endeqnarray\endsubequations
  }
```

`\@lab@subeqnarray` This macro picks up the `\label` command and its argument and re-inserts it *before* starting the eqnarray environment.

```
\newcommand*{\@lab@subeqnarray}[2]{#1{#2}\eqnarray}
```

⟨/package⟩

5 An example of the use of this package

When you run the following document through LaTeX you will see the differene between the `subeqnarray` and `eqnarray` environments.

```
⟨*sample⟩
\documentclass{article}
\usepackage{subeqn}

\begin{document}
This is an example ot the use of the \texttt{subeqations} package.
\begin{equation}
  \label{a}
  a^2 + b^2 = c^2
\end{equation}
Now we start sub-numbering.
\begin{subequations}
  \label{b}
  \begin{equation}
    \label{b1}
    d^2 + e^2 = f^2
  \end{equation}
  We can refer to equation~\ref{a}, \ref{b} and~\ref{b1}.
  \begin{equation}
    \label{b2}
    g^2 + h^2 = i^2
  \end{equation}
  This was equation~\ref{b2}.
  \begin{eqnarray}
    \label{c}
    x &=& y+z\label{c1}\\
    u &=& v+w\label{c2}
  \end{eqnarray}
  This was expression~\ref{c}, consisting of parts~\ref{c1}
  and~\ref{c2}.
\end{subequations}

Now lets start a \textsf{subeqnarray} environment.
\begin{subeqnarray}
  \label{d}
  x &=& y+z\label{d1}\\
  u &=& v+w\label{d2}
\end{subeqnarray}
This was equation~\ref{d}, with parts~\ref{d1} and~\ref{d2}.
\end{document}
⟨/sample⟩
```