LaTeX package eqnalign

— — making eqnarray(*) look and work like align(*) — —

by Tom Hejda, tohecz@gmail.com

1 The goal

The goal of this package is to allow easy conversion from the insanely-looking eqnarray to the look and behaviour of align from amsmath. It is inspired by a TeX.StackEchange question http://tex.stackexchange.com/q/96210/11002 by a user called “Werner”, and by an answer of mine to the question.

2 The behaviour

The package is activated by simply loading it, and it does not have any package options. It just redefines eqnarray and eqnarray*, and then it makes amsthm aware of this redefinition so that \qedhere works inside these environments.

3 License and remarks

(1) The package is licensed under the LaTeX Project Public License version 1.3c (LPPL v1.3c) or higher. The latest version of this license is in http://www.latex-project.org/lppl.txt.

(2) Note that actually, the usage of this package is discouraged, in favour of converting the code into proper “amsmath code”, using the true align and align* environments. It is intended for cases where a lot of already existing code needs to be converted and there is no capacity for doing it the right but time-consuming way.

(3) All bugs shall be reported to the GitHub page http://github.com/tohecz/eqnalign. Just note that unless the bug is crucial or easy to deal with, it may not be fixed since (per the previous remark) single problematic cases shall be dealt with by other means.

4 Implementation

Note that we in general say eqnarray where we really mean either eqnarray or eqnarray*.

```
⟨*package⟩
```

Package header.

```
\ProvidesPackage{eqnalign}[2016/06/09 v1.0 Make eqnarray(*) behave like align(*).]
```

The only necessary package is amsmath so that align and align* are defined.

```
\RequirePackage{amsmath}
```

The package does some catcode mysteries that shouldn't propagate out, so we make everything in a group and use \gdef everywhere.

```
\begingroup
```

\eqna@origamp We store a catcode-4 (tab alignment char) & in a macro. We need a catcode-13 (active) & througout the rest of the package.

```
\catcode`\&=4
\gdef\eqna@tab@amp{&}
\catcode`\&=13
```

\eqna@new@amp This will be the replacement of & inside eqnarray. We use \eqna@amp@ if the innermost environment is eqnarray and \eqna@origamp otherwise; this is to allow things like arrays and matrices inside eqnarray.

```
\gdef\eqna@new@amp{%
   \ifx\@currenvir\eqna@currenvir
      \eqna@amp@
   \else
      \eqna@tab@amp
   \fi
}
```

\eqna@amp@i
\eqna@amp@ii
\eqna@amp@iii Three macros that are "rotated", after the first, the second shall be used, then the third. The third one ends a group since it ends a table cell, therefore after that the first one is again in action. The first & on a line is kept, the second is ignored, the third is kept.

```
\gdef\eqna@amp@i{\eqna@tab@amp\let\eqna@amp@\eqna@amp@ii}
\gdef\eqna@amp@ii{\let\eqna@amp@\eqna@amp@iii}
\gdef\eqna@amp@iii{\eqna@tab@amp}
```

\eqna@doamp The default is \eqna@amp@i.

```
\global\let\eqna@amp@\eqna@amp@i
```

\eqna@hook The default meaning of & is \eqna@amp@i. We store the current environment, which is either eqnarray or eqnarray*; it is used in \eqna@new@amp for the test for inner environments. Then we activate & and make its meaning \eqna@new@amp.

```
\gdef\eqna@hook{%
   \let\eqna@currenvir\@currenvir
   \catcode`\&=\active
   \let&\eqna@new@amp
}
```

Now we will be defining environments containing * in name, so we make it a letter.

```
\catcode`\*=11
```

\eqna@def@env
eqnarray
eqnarray* We define a macro \eqna@def@env that contains the redefinitions of eqnarray (and eqnarray*). The environments themselves are like align, just hooked using \eqna@hook. We then call this macro immediately to define the environments. (All this fuss with \eqna@def@env is to correct things in case hyperref is loaded after eqnalign.

```
\gdef\eqna@def@env{%
   \gdef\eqnarray{\eqna@hook\align}%
   \gdef\eqnarray*{\eqna@hook\align*}%
   \global\let\endeqnarray\endalign
   \global\let\endeqnarray*\endalign*
}
\eqna@def@env
```

\eqnarray@qed
\eqnarray*@qed To make amsthm's \qedhere work in eqnarray, we need to "hint" amsthm that it exists

```
\global\let\eqnarray@qed\align@qed
\global\let\eqnarray*@qed\align*@qed
```

End the group we began at the very beginning.

```
\endgroup
```

EQNarray Just of sentiment, we keep the original eqnarray as EQNarray.

```
\def\EQNarray{%
   \stepcounter{equation}%
   \def\@currentlabel{\p@equation\theequation}%
   \global\@eqnswtrue
   \m@th
   \global\@eqcnt\z@
   \tabskip\@centering
   \let\\\@eqncr
   $$\everycr{}\halign to\displaywidth\bgroup
       \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnsel
      &\global\@eqcnt\@ne\hskip \tw@\arraycolsep \hfil${##}$\hfil
      &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
         $\displaystyle{##}$\hfil\tabskip\@centering
      &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
         \tabskip\z@skip
      \cr
}
\def\endEQNarray{%
      \@@eqncr
      \egroup
      \global\advance\c@equation\m@ne
   $$\@ignoretrue
}
\@namedef{EQNarray*}{\def\@eqncr{\nonumber\@seqncr}\EQNarray}
\@namedef{endEQNarray*}{\nonumber\endEQNarray}
```

Last but not least, if hyperref is loaded after eqnalign (and only in that case), we issue a warning since hyperref is doing bad things to eqnarray, and we redefine eqnalign once more.

```
\@ifpackageloaded{hyperref}{}{
   \AtBeginDocument{
      \@ifpackageloaded{hyperref}{
         \@latex@warning{Package ‘eqnalign’ should be loaded after ‘hyperref’.\MessageBreak
         Redefining ‘eqnarray’ and ‘eqnarray*’ at this point \MessageBreak and crossing fin
         \eqna@def@env
      }{}
   }
}
```

That’s all.

```
\endinput
```

⟨/package⟩