The \texttt{nccmath} package*

Alexander I. Rozhenko
rozhenko@oapmg.sscc.ru
2006/01/20

The package extends the \texttt{amsmath} package adding some math commands from NCC-L\TeX. It also improves spacing control before display equations and fixes a bug of ignoring the \texttt{\displaybreak} in the \texttt{amsmath} version of the \texttt{equation} environment. All options are passed to the \texttt{amsmath} package.

Contents

1 Improvement to the \texttt{amsmath} 1
2 Extra Macros 2
3 Medium-Size Math Commands 3
4 NCC-\texttt{\LaTeX} Equivalents to Display Formulas 5
5 The Implementation 5
  5.1 Kernel 6
  5.2 Additional Math Commands 6
  5.3 Medium-Size Math Commands 7
  5.4 Patches to \texttt{amsmath} 11
  5.5 The \texttt{darray} Environment 14
  5.6 NCC Equations 16
  5.7 Math with medium fractions and operators 18

1 Improvement to the \texttt{amsmath}

eqnarray In the \texttt{amsmath} package, the \texttt{eqnarray} environment leaves unchanged because alternative \texttt{\LaTeX}s environments exist. We redefine the \texttt{eqnarray} to work in the \texttt{\LaTeX}s style. The following improvements are done in it: an equation tag is prepared by the same manner as in \texttt{\LaTeX}s display formulas (\texttt{\tag} and \texttt{\tag*} are allowed); the \texttt{\displaybreak} command is allowed; the intercolumn distance is reduced to

\*This file has version number v1.3, last revised 2006/01/20.
the distance between ordinary and relational math symbols; and the center field
is prepared in the `\textstyle` (the original version uses `\displaystyle` here).

The `\intertext` command is improved here. It now has an optional parameter:

\begin{verbatim}
\intertext[distance]{text}
\end{verbatim}

The \textit{distance} parameter specifies a vertical space inserted before and after the
text. If it is omitted, standard \TeX's skips are inserted.

The following changes are made in display equations:

- The `\displaybreak` command now works within the \textit{equation} environment
  (it is ignored in the \texttt{amsmath});
- The \texttt{AMS} and \LaTeX display equations prepared in the vertical mode do not
  produce now an empty extra line before. Moreover, if a minipage starts from
  a display formula, the vertical skip before is suppressed.

2 Extra Macros

\texttt{fleqn} \texttt{ceqn} The following environments allow change the horizontal alignment of formulas
inside them:

\begin{verbatim}
\begin{fleqn}[\textit{margin}] \ldots \end{fleqn}
\begin{ceqn} \ldots \end{ceqn}
\end{verbatim}

The \texttt{fleqn} environment prepares inner display equations in the flush left style.
The \textit{margin} parameter specifies the left margin value. If it is omitted, zero
value is used. The \texttt{ceqn} environment prepares inner display equations in the
centered style. They have no effect on formulas prepared with the low-level \TeX
command $$.\$

\texttt{darray} The \texttt{darray} environment produces an array of formulas in the \texttt{displaystyle}.
The distance between formulas is enlarged in just the same manner as in other
d multiline display equations. The \texttt{darray} environment has the same syntax as the
\texttt{array}:

\begin{verbatim}
\begin{darray}[\textit{pos}]{\textit{columns}}
\langle body\rangle
\end{darray}
\end{verbatim}

The \textit{pos} argument describes the vertical alignment of the array box (t, b, or
c; default is c). The use of column specifications in the \textit{columns} argument is
restricted in comparison with \texttt{array}: it can contain the 1, c, and r specifiers, * and @ commands. The intercolumn separation is smaller than in the \texttt{array}: it is
reduced to the distance between ordinary and relational math symbols. As in the
\texttt{amsmath} package, the thin skip is inserted before \texttt{darray}. Skips before the first
and after the last column of \texttt{darray} are not inserted. To insert them manually,
use $\&\ldots$ in the \textit{columns} argument.

The \texttt{darray} environment is implemented independently on the \texttt{array} environment to avoid conflicts with the \texttt{array} package.
\multicolumn{1}{c}{\text{\textbackslash{multicolumn}}(\textit{count})\{\textit{preamble}\}\{\textit{formula}\}} is used in \texttt{darray} instead of \texttt{multicolumn}.

\texttt{\textbackslash{useshortskip}} In \LaTeX{}, two types of skips above display formulas are used: the normal skip defined in the \texttt{\textbackslash{abovedisplayskip}} register and the short skip defined in the \texttt{\textbackslash{abovedisplayshortskip}} register. When a display formula is typed out, \LaTeX{} decides what skip to insert depending on the width of formula, its style (centered or flushed left, numbered left or right), and the width of the rest of text in the last line of the previous paragraph. But this algorithm works for ordinary formulas only. It does not work in multiline formulas prepared with \texttt{\textbackslash{halign}} command. So, a manual replacement of the normal skip to the short skip is required in some cases. To provides this, the \texttt{\textbackslash{useshortskip}} command is introduced. It forces the use of short skip in the next display formula but it has no effect on formulas prepared with the low-level \LaTeX{} command \texttt{\textbackslash{}$}. The vertical distance between lines of multiline equations is frequently smaller than necessary. To increase it, the extra distance can be used as the optional parameter of the \texttt{\textbackslash{}[\textit{dist}] command. In most cases, it is enough to increase the distance on 0.5ex. We introduce the \texttt{\textbackslash{nr}} command here that is equivalent to the \texttt{\textbackslash{}[0.5ex]}. Its full sintax is just the same as for the \texttt{\textbackslash{}} command:

\texttt{\textbackslash{nr}\{\textit{dist}\}}

This command can be used everywhere the command \texttt{\textbackslash{}} is allowed.

\texttt{\textbackslash{mrel}} The \texttt{\textbackslash{mrel}\{\textit{column}\}} command composes a new math relation symbol from a one-column stack of math formulas described in the \texttt{\textit{column}} parameter. For example, the command \texttt{$\textbackslash{mrel}\{<\textbackslash{-0.7ex}>\}$} produces $\leq$.

\texttt{\textbackslash{underrel}} The \texttt{\textbackslash{underrel}\{\textit{base}\}\{\textit{bottom}\}} command is a twin to the \texttt{\textbackslash{overrel}} command. For example, the command \texttt{$A\textbackslash{underrel}\{\longrightarrow\}\{x\to 0\}B$} produces $A \rightarrow_{x \to 0} B$.

### 3 Medium-Size Math Commands

Since version 1.2, a collection of medium-size math commands is introduced.

\texttt{\textbackslash{medmath}} The \texttt{\textbackslash{medmath}\{\textit{formula}\}} command decreases a size of formula in 1.2 times and prepares it in the display style. An example:

\begin{verbatim}
$\textbackslash{medmath}\{\cfrac{1}{\sqrt 2 +\cfrac{1}{\sqrt 2 +\dotsb}}\}$\quad $\cfrac{1}{\sqrt 2 +\cfrac{1}{\sqrt 2 +\dotsb}}$
\end{verbatim}

It produces:

\[
\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \cdots}} \quad \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \cdots}}
\]

\texttt{\textbackslash{medop}} The \texttt{\textbackslash{medop}\{\textit{operator}\}} command prepares a medium-size operator with the required preference for limits. It can be use with \texttt{\textbackslash{sum}} and others variable-size commands except integrals. An example:

\begin{verbatim}
3
\end{verbatim}
$$\sum_{i=1}^n \medint\sum_{i=1}^n \displaystyle \sum\nolimits_{i=1}^n$$

It produces:

\[
\sum_{i=1}^n \sum_{i=1}^n \sum_{i=1}^n
\]

\texttt{\medint} \quad \text{The} \ \texttt{\medint\{\langle\text{operator}\rangle\}} \quad \text{command prepares a medium-size integral with required preference for limits. It can be use with} \ \texttt{\int\text{-family}} \quad \text{of commands and} \ \texttt{\oint} \text{command. An example:}

\[
\int_a^b \medint\int_a^b \displaystyle\int_a^b
\]

\[
\int\limits_a^b \medint\int_a^b\limits \displaystyle\int_a\limits^b
\]

\[
\iint_a^b \medint\iiint_a^b \displaystyle\iiiint_a^b
\]

\[
\iint\limits_X^Y \medint\iiint_X\limits^Y \displaystyle\iiiint_X^Y\limits
\]

\[
\medint\idotsint_X\limits \medint\oint_X^Y
\]

\text{By the way, the original limits recognizing in} \ \texttt{amsmath} \quad \text{multi-integrals is very restrictive: it allows only one} \ \texttt{\limits}\text{-like command right after the multi-integral. In this package, the recognizing is improved to work as} \ \texttt{\TeX}'s \text{one.}

\texttt{\medintcorr\{\langle\text{length}\rangle\}} \text{command specifies the value of italic correction for medium integrals. It controls a positioning indices in medium integrals and in multi-integrals. Its default value is} \ 0.5 \text{em.}

\texttt{\mfrac} \quad \text{Based on the medium size formulas, the} \ \texttt{\mfrac} \quad \text{and} \ \texttt{\mbinom} \quad \text{commands are introduced. They are similar to} \ \texttt{\frac} \quad \text{and} \ \texttt{\binom}. \ \text{An example:}

\[
\frac {x+y}{a-b} \mfrac {x+y}{a-b} \dfrac {x+y}{a-b}
\]

\[
\binom {n}{k} \mbinom {n}{k} \dbinom {n}{k}
\]

\texttt{\medsize} \quad \text{The} \ \texttt{medsize} \quad \text{environment is introduced to prepare formulas and arrays in the medium size. It reduces the} \ \texttt{\arraycolsep} \quad \text{value by 0.8 times. Basing on it, the} \ \texttt{mmatrix} \quad \text{environment is introduced. It is specified as follows:}

\[
\begin{mmatrix} ... \end{mmatrix} \equiv \begin{medsize}\begin{matrix} ... \end{matrix}\end{medsize}
\]

\text{An example:}
It produces:
\[
\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix}
\]

Finally, the \texttt{mediummath} option allows prepare all variable-size math elements in medium size. It redefines \texttt{\frac}, \texttt{\binom} and all math operators to the medium size. For \texttt{\frac} and \texttt{\binom}, the medium size is applied in the display and text styles. The \texttt{\dfrac}, \texttt{\tfrac}, \texttt{\dbinom}, and \texttt{\tbinom} commands have the old meaning.

\section{NCC-\LaTeX Equivalents to Display Formulas}

The following NCC-\LaTeX equivalents are provided with this package:

\begin{verbatim}
\eq{formula}  = \begin{equation} \text{formula} \end{equation}.
\eq*{formula} = \begin{equation*} \text{formula} \end{equation*}.
\eqs{formulas} = \begin{eqnarray} \text{formulas} \end{eqnarray}.
\eqs*{formulas} = \begin{eqnarray*} \text{formulas} \end{eqnarray*}.
\eqalign{formulas} = \begin{equation} \begin{darray}{rcl} \text{formulas} \end{darray} \end{equation}.
\eqalign*{formulas} = \begin{equation*} \begin{darray}{rcl} \text{formulas} \end{darray} \end{equation*}.
\end{verbatim}

The \texttt{\eqs} and \texttt{\eqs*} commands have an optional parameter specifying a distance between columns. For example, in the command

\begin{verbatim}
\eqs[0mm]{&& -\Delta u = f, \ \& \ \& \ |u|_{\Gamma} = 0},
\end{verbatim}

the intercolumn distance is removed because only the 3rd column is used. The \texttt{eqnarray} environment has no optional parameter.

The \texttt{\eqalign} and \texttt{\eqalign*} commands also have an optional parameter. Its meaning is the column specification parameter: \texttt{\eqalign{formulas} = \eqalign[rcl]{formulas}}.

\section{The Implementation}

At first we load the \texttt{amsmath} package and pass all options to it except the \texttt{mediummath} option.

\footnotesize 1 (package)
5.1 Kernel

\NCC@cr
Simplified version of $\textbackslash$ used in some commands here. The low level command
\NCC@cr@@(@skip@) is defined if necessary to \NCC@aligncr or to something else.
The \texttt{new@ifnextchar} commands from the \texttt{AMS} does the same as \texttt{ifnextchar},
but disallows spaces before the tested symbol.

6 newif\ifNCC@star
7 \def\NCC@cr{\relax\iffalse{\fi\ifnum0='}\fi
8 \@ifstar{\global\NCC@startrue\NCC@cr@}{\global\NCC@starfalse\NCC@cr@}%
9 }
10 \def\NCC@cr@{\new@ifnextchar\[\NCC@cr@@{@NCC@cr@@@{\[\]}
11 \def\NCC@cr@@@[#1]{\ifnum0='{\fi \iffalse}\fi\NCC@cr@@@{#1}}
12 \def\NCC@aligncr#1{\cr\noalign{\vskip #1\relax}}
13 \NCC@default@cr
This command sets defaults for the $\textbackslash$ command.
14 \NCC@default@cr
15 \let\NCC@cr\relax
16 \let\NCC@cr@@@\NCC@aligncr
17 \NCC@cr
18 }
19 \def\NCC@nr#1{%
20 \let\NCC@temp\NCC@cr@@@
21 \let\NCC@cr@@@\NCC@nr
22 \NCC@cr
23 }
24 \def\NCC@nr#1{%
25 \let\NCC@temp\NCC@cr@@@
26 \ifNCC@star
27 \edef\@tempa{\noexpand\[
28 }
29 \else
30 \edef\@tempa{\noexpand\[
31 \fi
32 \@tempa
33}

5.2 Additional Math Commands

\mrel
The \texttt{\mrel\textbackslash{}column\textbackslash{}} command composes a new math relation and vertically
centers it with respect to the math line.

29 \newcommand{\mrel}{\mathpalette\NCC@rel}
30 \def\NCC@rel#1#2{\mathrel{\vcenter{\NCC@default@cr
31 \offinterlineskip \ialign{\m@th#1##\m@th#2\crcr}}}
The \underrel{⟨base⟩}{⟨bottom⟩} command is a twin to \overrel.

\newcommand{\underrel}[2]{\mathrel{\mathop{#1}\limits_{#2}}}

5.3 Medium-Size Math Commands

The \NCC@select@msize command prepares dimensions for medium-size math:

- In \NCC@fracrulewidth — a rule width in fractions;
- In @tempdima — a raising value; and
- In @tempdimb — a font size to be used in medium fractions and matrices.

\NCC@innerfrac{⟨style⟩} prepares a fraction with a special width in the given style:

\NCC@prepare@msize Select a font by rounding its pt-size to the nearest integer and redefine fractions to have the given rule width. The \binom command is redefined also to its original value because it can be changed when the mediummath option is applied.
\NCC@op\prepare \NCC@op\prepare\text{\textup{(integral)}} command prepares an integral. It looks forward, extracts indices and limits-change commands, and puts the integral with required kerning of indices. The \NCC@op\print driver is a command to print the integral. Its default value is \NCC@op\printm. The driver uses the following hooks: \NCC@op contains an integral command, \NCC@op\lim contains the selected limits-style, \NCC@op\sb contains a subscript, \NCC@op\sp contains a superscript, \NCC@op\kern contains the kerning value for medium-size integrals. If subscript or superscript is omitted, the corresponding hook is equal to \relax.

\DeclareRobustCommand*\NCC@op\prepare[1]{% 
  \def\NCC@op{#1}\
  \let\NCC@op\print\NCC@op\printm 
  \NCC@op\prepare@ 
}

\def\NCC@op\prepare@[1]{% 
  \let\NCC@op\lim\ilimits@ 
  \let\NCC@op\sp\relax 
  \let\NCC@op\sb\relax 
  \NCC@op\next 
}

\def\NCC@op\next{\futurelet\@let@token\NCC@op\getnext} 
Test the next token and get it if necessary: 
\def\NCC@op\getnext[1]{% 
  \let\@tempa\NCC@op\skip 
  \ifx\@let@token\limits 
    \let\NCC@op\lim\limits 
  \else 
    \ifx\@let@token\nolimits 
      \let\NCC@op\lim\nolimits 
    \else 
      \ifx\@let@token\displaylimits 
        \let\NCC@op\lim\displaylimits 
      \else 
        \ifx\@let@token\sp 
          \NCC@op\test\NCC@op\sp 
        \else 
          \def\@tempa{\NCC@op\get\NCC@op\sp}\else 
        \fi 
      \fi 
    \fi 
  \fi 
  \NCC@op\test\NCC@op\sb 
  \def\@tempa{\NCC@op\get\NCC@op\sb}\else 
  \ifx\@let@token\sb 
    \NCC@op\test\NCC@op\sb 
  \else 
    \def\@tempa{\NCC@op\get\NCC@op\sb}\else 
  \fi 
  \NCC@op\test\NCC@op\sp 
  \let\@tempa\NCC@op\skipsp \else 
  \let\@tempa\NCC@op\print 
  \fi 
}\fi
95 \fi
96 \fi
97 \fi
98 \@tempa
99 }

Skip \limits-like token:
100 \def\NCC@op@skip1{\NCC@op@next}
101 \def\NCC@op@skip2{\NCC@op@next}
102 \def\NCC@op@skip3{\NCC@op@next}
103 }

Skip a space token. A space token is skipped within \@ifnextchar before comparing it with the first parameter. So, it does not important what char to test for:
104 \def\NCC@op@skipsp{\@ifnextchar0{\NCC@op@next}{\NCC@op@next}}
105 \def\NCC@op@test#1{\ifx#1\relax \PackageError{nccmath}{Double index in math operator}{}
106 \fi}
107 \def\NCC@op@get#1#2#3{\def#1{#3}\NCC@op@next}
108 \NCC@op@printm

Driver for printing the medium-size integral with indices:
109 \def\NCC@op@printm{\NCC@op@printm@\NCC@op\NCC@op@kern}
110 \def\NCC@op@printm@{\NCC@op@print@\NCC@op@print@{⟨integral⟩}{⟨kern⟩}{⟨level⟩} command prints an ⟨integral⟩ using the specified ⟨kern⟩ in indices. If ⟨level⟩ = 0 use \limits else use \nolimits.
111 \def\NCC@op@print@#1#2#3{\mathop{#1}{\setlength\@tempdima{#2}{\ifx\NCC@op@sb\relax \else \ifnum#3>\z@ \@tempswafalse \fi \fi\ifx\NCC@op@sp\relax \else \ifnum#3>\z@ \@tempswafalse \fi \fi\edef\@tempa{\ifnum#3=\z@ \noexpand\limits \else \noexpand\nolimits \fi}{\ifnum#3=\z@ \noexpand\sb{\kern -\@tempdima\else \kern -.8\@tempdima \fi}}}}
\medmath The \medmath{\langle formula \rangle} prepares a medium-size formula in display style:
\medop The \medop{\langle operator \rangle} prepares an operator in the medium size:
\medintcorr The \medintcorr{\langle length \rangle} specifies an italic correction for a medium integral:
\medint The \medint{\langle integral \rangle} command prepares a medium integral:
\mfrac The \mfrac{\langle numerator \rangle}{\langle denominator \rangle} prepares a medium-size fraction:
\mbinom The \mbinom{\langle numerator \rangle}{\langle denominator \rangle} prepares a medium-size binomial expression:
medsize The medsize environment is useful for preparing medium-size arrays:
mmatrix The mmatrix environment prepares a medium-size matrix:
5.4 Patches to amsmath

\MultiIntegral Improve the MultiIntegral kerning method on the base of \NCC@op@prepare@ hook. The original method from amsmath works bad if a multi-integral is an argument of the \medint command.

\begin{verbatim}
\renewcommand*{\MultiIntegral}{% 
  \edef\NCC@op{\noexpand\intop} 
  \ifnum#1=\z@\noexpand\intdots@ \else\noexpand\intkern@ \fi 
  \ifnum#1>\tw@ \noexpand\intop \noexpand\intkern@ \fi 
  \ifnum#1>\thr@@ \noexpand\intop \noexpand\intkern@ \fi 
  \noexpand\intop} \let\NCC@op@print\NCC@op@printd \NCC@op@prepare@ 
\end{verbatim}

\begin{verbatim}
\def\NCC@op@printd{% 
  \setlength{\@tempdima}{\NCC@op@kern} 
  \ifx\NCC@op@lim\nolimits \@tempcnta\@ne \else \ifx\NCC@op@lim\limits \@tempcnta\z@ \else \@tempcnta\m@ne \fi \fi 
  \mathchoice{\NCC@op@printd@{\displaystyle}{1.2\@tempdima}}% 
  {\NCC@op@printd@{\textstyle}{.8\@tempdima}}% 
  {\NCC@op@printd@{\scriptstyle}{.8\@tempdima}}% 
  {\NCC@op@printd@{\scriptscriptstyle}{.8\@tempdima}}% 
\end{verbatim}

\begin{verbatim}
\def\NCC@op@printd@#1#2{#1 \ifnum\@tempcnta>\m@ne \NCC@op@print@{\hbox{$#1\NCC@op$}}{#2}\@tempcnta \else \ifx#1\displaystyle \NCC@op@print@{\hbox{$#1\NCC@op$}}{#2}\z@ \else \NCC@op@print@{\hbox{$#1\NCC@op$}}{#2}\@ne \fi \fi} \end{verbatim}

\endmathdisplay@a Fix the bug in the \endmathdisplay@a command from the amsmath package. The \displaybreak has no effect in it if a tag is specified. This is because the change of \postdisplaypenalty is done after the \eqno command. But the rest of display formula after \eqno up to the $$ command belongs to the tag. It is prepared in the horizontal mode and the mentioned penalty is ignored. Fixed version of this command at first changes the \postdisplaypenalty and after that prints a tag. To be sure, that the required command does not fixed yet, we prepare its bug version in the \@tempa command.

\begin{verbatim}
\end{verbatim}

\begin{verbatim}
\def\@tempa{% 
  \if@eqnsw \gdef\df@tag{\tagform@\theequation}\fi} \end{verbatim}

\endmathdisplay@a
and compare it with the current value of \endmathdisplay@a. If they are identic, we fix the last command. Otherwise, print a warning and do nothing.

\ifx\@tempa\endmathdisplay@a
\def\endmathdisplay@a{\
\ifnum\dspbrk@lvl>\m@ne
\postdisplaypenalty -\@getpen\dspbrk@lvl\
global\dspbrk@lvl\m@ne
\fi
\if@eqnsw \gdef\df@tag{\tagform@\theequation}\fi
\if@fleqn \@xp\endmathdisplay@fleqn
\else \ifx\df@tag\@empty \else \veqno \alt@tag \df@tag \fi
\ifx\df@label\@empty \else \@xp\ltx@label\@xp{\df@label}\fi
\fi
\else
\PackageWarning{nccmath}\
{The \string\endmathdisplay@a\ command differs from amsmath package. We don't fix it!}
\fi
\fi
\intertext
Redefine \textit{AMS}’s \texttt{\intertext{(text)}} to \texttt{\intertext[(skip)](text)}. Optional \texttt{(skip)} means the vertical space inserted below and after the text. If it is omitted, the default \texttt{belowdisplayskip} and \texttt{abovedisplayskip} spaces are inserted.

We need to redefine its default value used out of display equations:

\renewcommand*{\intertext}{\texttt[]}
\texttt{\intertext[(skip)](text)}
and also must redefine the \texttt{\intertext@} hook that changes the value of \texttt{\intertext} within display equations. Its new definition differs from the original one in the conditional inserting of skips before and after the text. The optional parameter is scanned inside the \texttt{\noalign} command. We use the ordinary trick with the \texttt{\ifnum0} to close the open brace in the next macro.

\def\intertext@{\def\intertext{\ifvmode\else\\@empty\fi
\noalign{\ifnum0='}\fi
\@ifnextchar[{{\NCC@intertext}{\NCC@intertext\[\]}}
\def\NCC@intertext[#1]{\NCC@intertext{}\NCC@intertext[]}%
}
\def\NCC@intertext[#1]{\NCC@intertext}
\useshortskip  The \useshortskip command changes an above skip for nearest display formula to \abovedisplayskips. Really, it sets the value of inner if-macro to true and the actual changes are applied in the \NCC@ignorepar hook.

\NCCignorepar  This command removes extra vertical space before display formula if it starts from a new paragraph and changes the before-skip to \abovedisplayskips if the \useshortskip command was applied.

\def\NCCignorepar{\def\NCCignorepar{% 
  \if\NCCshortskip\else\fi 
  \global\NCCshortskipfalse
  \ifnolinebreak\else\fi 
  \ifnolinebreakfalse\else\fi 

  \addvspace{\abovedisplayskips}\z@skip
  \abovedisplayskips\z@skip
  \abovedisplayskips\z@skip
  \belowdisplayskips\z@skip
  \noindent
  \fi
  \fi
}

Now we insert the \NCCignorepar command at the beginning of all \LaTeX and \ams-L\LaTeX display equations except eqnarray. We need to correct four \ams commands only:

\let\NCCstartgather\start@gather
\let\NCCstartalign\start@align
\let\NCCstartmultline\start@multline
\let\NCCstartdisplay\start@display
\def\start@gather{\NCCignorepar\NCCstartgather}
\def\start@align{\if@gather\NCCignorepar\fi\NCCstartalign}
5.5 The darray Environment

darray

The implementation of darray is a hybrid of the \start@aligned command from the amsmath package and the \array command.

\newenvironment{darray}{% #1 #2 #3}{% \null % \if #1t \vtop \else \if#1b \vbox \else \vcenter \fi \fi % \bgroup % \NCC@default@cr % \spread@equation % \NCC@mkpream{#2}% % \edef\@preamble{\ialign \bgroup \strut@ \@preamble \tabskip\z@skip \cr} % \let\par\@empty % \let\@sharp##% % \set@typeset@protect % \tabskip\z@skip % \@preamble % \endgroup % \ignorespaces % \cr % % \crcr \egroup \egroup %}

dmulticolumn

To produce multi-columns in darray, the dmulticolumn command is used.

\newcommand\dmulticolumn[3]{% \multispan{#1}% \begingroup % \NCC@mkpream{#2}% % \def\@sharp{#3}\set@typeset@protect % \@preamble % \endgroup % % \ignorespaces %}

NCC@mkpream

The darray environment is independent from array to avoid conflicts with packages customizing the array environment. So, we need to implement an independent preamble maker.

The following classes can appear in the preamble:

0 1cr 1 @-argument 2 @

The implementation of preamble maker is very similar to the \LaTeX's version.

\def\NCC@mkpream#1{% \@lastchclass\@ne \@firstamptrue % \set@typeset@protect % % \set@typeset@protect % % \multispan{#1}% \set@typeset@protect % % \multispan{#1}% % % \@preamble % % \endgroup % % \ignorespaces % % \cr % % \cr % % \crcr % % \egroup % % \egroup % %}

\settowidth\alignsep$\m@th\mskip\thickmuskip$%
The \@xexpast command expands the argument replacing all instances of *{(N)}*{(string)} by (N) copies of (string). The result is saved in the \reserved@a macro. But this command is let to \relax in the array package. So, we use its original definition prepared in the \NCC@xexpast macro to avoid conflicts with other packages.

Now we make the preamble collecting it in the \@preamble hook. The code is very similar to the \LaTeX’s \@xexpast command.

\NCC@xexpast

The standard \LaTeX’s \@xexpast macro is saved here:

\def\NCC@xexpast#1*#2#3#4\@@{
def\reserved@a{#1}
5.6 NCC Equations

fleqn  The implementation of these environments is straightforward: change the \if@fleqn flag and the \@mathmargin value:

\newenvironment{fleqn}{\@fleqntrue}{%}
\newenvironment*{fleqn}{\@fleqntrue}{%}
\newenvironment{ceqn}{\@fleqntfale}{%}
\newenvironment*{ceqn}{\@fleqntfale}{%}

\eq The implementation of the NCC-L\TeX's \eq command is quite simple:

\newcommand{\eq}{\@ifstar{\NCC@eqx}{\NCC@eq}}
\def{\NCC@eqx}{\begin{equation*}{#1}\end{equation*}}
\def{\NCC@eq}{\begin{equation}{#1}\end{equation}}

\eqalign The \eqalign command is based on the equation and darray environments:

\newcommand{\eqalign}{\@ifstar{\let{\@tempa}{\NCC@eqx} \NCC@eqa}{\let{\@tempa}{\NCC@eq} \NCC@eqa}}
\newcommand*{\NCC@eqa}{\begin{darray}{#1}{#2}\end{darray}}

\eqs The difference between the \eqs command and the eqnarray environment consists in optional length parameters allowed in \eqs. All these commands are based on \NCC@beqs and \NCC@eeqs macros.

\newcommand{\eqs}{\@ifstar{\st@rredtrue \NCC@eqs}{\st@rredfalse \NCC@eqs}}
\newcommand*{\NCC@eqs}{\begin{group}\NCC@beqs{#1}{#2}\NCC@eeqs\end{group}}
\renewenvironment{eqnarray}{\st@rredfalse \NCC@beqs{}}
The \texttt{\NCC@beqs\{\textit{skip}\}} starts eqnarray-like equations. The \textit{\textlangle{\textit{skip}}\textrangle} parameter specifies a skip inserted between columns. If it is empty, the default value of this skip is used. It equals to the thick skip appearing in relations. The implementation of this macro uses hooks from the \texttt{amsmath} package.

\def\NCC@beqs#1{%}
  \NCC@eqcr
  \ifst@rred\else
    \global\@eqnswtrue
  \fi
  \\cr
\def\NCC@eqcr{\
  \ifcase\column@ \def\@tempa{&&&} \or \def\@tempa{&&} \or \def\@tempa{&} \else \let\@tempa\@empty \@latex@error{Too many columns in eqnarray environment}\@ehc
  \fi
  \ifst@rred\nonumber\fi
  \if@eqnsw \global\tag@true \fi
  \iftagsleft@ \rlap{\hskip -\displaywidth\make@display@tag} \else \make@display@tag \fi
  \fi
  \ifst@rred\else\global\@eqnswtrue\fi
  \cr
}

This macro finishes eqnarray-like equations.

\def\NCC@eeqs{\math@cr\egroup\$}
5.7 Math with medium fractions and operators

Finally, we process the mediummath option. It is recognized by the \NCC@op command to be specified.
402 \ifundefined{NCC@op}{\endinput}{}

Redefine fractions and binoms.
403 \DeclareRobustCommand\frac{\NCC@op@select\mfrac{\genfrac{}{}{}{}}}
404 \DeclareRobustCommand\binom{\NCC@op@select\mbinom{\genfrac(){}}}
405 \def\NCC@op@select#1#2#3#4{%}
406 \mathchoice{#1#3#4}{\scriptstyle#2#3#4}{\scriptscriptstyle#2#3#4}{%}
407 {\scriptstyle#3#4}{\scriptstyle#3#4}%
408 }

Redefine all math operators except integrals:
409 \def\@tempa#1#2{%}
410 \ifx#2\@undefined \let#2#1\fi
411 \def#1{\DOTSB\medop{#2}}%
412 }
413 \@tempa \coprod \coprod@
414 \@tempa \bigvee \bigvee@
415 \@tempa \bigwedge \bigwedge@
416 \@tempa \biguplus \biguplus@
417 \@tempa \bigcap \bigcap@
418 \@tempa \bigcup \bigcup@
419 \@tempa \prod \prod@
420 \@tempa \sum \sum@
421 \@tempa \bigotimes \bigotimes@
422 \@tempa \bigoplus \bigoplus@
423 \@tempa \bigodot \bigodot@
424 \@tempa \bigsqcup \bigsqcup@

Redefine integrals:
425 \def\@tempa#1#2#3{
426 \DeclareRobustCommand#2{\mathop{\medmath{#3}}}%
427 \def#1{\DOTSI\NCC@op@prepare{#2}}%
428 }
429 \@tempa\int \intop \NCC@op@int
430 \@tempa\oint \ointop \NCC@op@oint
431 \let\@tempa\relax

Redefine multiple integrals:
432 \renewcommand*[\MultiIntegral]{1}{%}
433 \def\NCC@op\noexpand\intop
434 \ifnum#1=\z@\noexpand\intdots@\else\noexpand\intkern0\fi
435 \ifnum#1>\tw@\noexpand\intop\noexpand\intkern0\fi
436 \ifnum#1>\thr@@\noexpand\intop\noexpand\intkern0\fi
437 \noexpand\intop
438 }%
439 \let\NCC@op\print\NCC@op\printm
440 \NCC@op\prepare0
\def\intkern0{\kern-\NCC@op@kern}
\def\intdots0{\setlength\@tempdima{\NCC@op@kern}\kern-.4\@tempdima{\cdotp}\mkern1.5mu{\cdotp}\mkern1.5mu{\cdotp}\kern-.4\@tempdima}
\langle\/package\rangle