The **autobreak** package*

Takahiro Ueda

23 February 2017

Abstract

This package implements a simple mechanism of line/page breaking within the `align` environment of the `amsmath` package; new line characters are considered as possible candidates for the breaks and the package tries to put breaks at adequate places. It is suitable for computer-generated long formulae with many terms.

Contents

1 Introduction 1
2 Usage 2
3 Caveats 6
4 Implementation 8
   4.1 Registers and constants . . . . . . . . . . . . . . . . . . . . 8
   4.2 Interaction with .aux files . . . . . . . . . . . . . . . . . . . 9
   4.3 Hacking amsmath . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
   4.4 autobreak environment . . . . . . . . . . . . . . . . . . . . . . . 12

1 Introduction

Sometimes people want to put long formulae in their documents, which do not fit in a line and may span over multiple pages. The following is an equation of explicitly writing down the first 50 terms in the sum of the well-known Basel problem:

\[
\zeta(2) = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \frac{1}{36} + \frac{1}{49} + \frac{1}{64} + \frac{1}{81} + \frac{1}{100} + \frac{1}{121} + \frac{1}{144} + \frac{1}{169} + \frac{1}{196} + \frac{1}{225} + \frac{1}{256} + \frac{1}{289} + \frac{1}{324} + \frac{1}{361} + \frac{1}{400} + \frac{1}{441} + \frac{1}{484} + \frac{1}{529}
\]

*This document corresponds to `autobreak` v0.3, dated 2017/02/23.
The above example might seem nonsense, but putting long formulae may have a meaning in some cases and become inevitable for completeness of documents, writing self-contained papers, or just to impress readers. They are typically generated as outputs of computer algebra systems, and would have the form of a sum of many terms while each term is short.

Then, the question is how to break long formulae in such a way that the expressions do not make any overfull lines for \LaTeX. Certainly, one can attempt to manually insert line breaks by trial and error, checking whether \LaTeX warns overfull lines, and this process could be automatized by external scripts at some extent. A shortcoming of such ‘manual’ approaches is that line breaks have to be reexamined whenever the layout of the document is changed, e.g., replacing the document class or reusing existing equations into another document with a different format.

The goal of the \texttt{autobreak} package is to give a reasonably simple solution for (semi-)automatic line breaking of long formulae within \LaTeX\textsuperscript{1}.

2 Usage

The \texttt{autobreak} package is supposed to be used together with the \texttt{amsmath} package:\textsuperscript{2,3}

\begin{verbatim}
\usepackage{amsmath}
\usepackage{autobreak}
\end{verbatim}

When your document contains long equations over multiple pages, you might want to use \texttt{allowdisplaybreaks} of \texttt{amsmath} package:

\begin{verbatim}
\allowdisplaybreaks
\end{verbatim}

\begin{verbatim}
\begin{autobreak}
⟨long-equations⟩
\end{autobreak}
\end{verbatim}

\textsuperscript{1}There is another package \texttt{breqn} (https://www.ctan.org/pkg/breqn), which adopts a more automatic fashion and is useful for more sophisticated line breaking, unless you get “Dimension too large” error for really big expressions.

\textsuperscript{2}https://www.ctan.org/pkg/amsmath.

\textsuperscript{3}Actually \texttt{autobreak} internally loads \texttt{amsmath}, but it is still a good practice to explicitly include all packages providing macros used in your document.
The \texttt{autobreak} environment is used for breaking lines in long formulae in the \texttt{align} environment of \texttt{amsmath}.

\begin{align}
\begin{autobreak}
\zeta(3) = & 1 + \frac{1}{8} + \frac{1}{27} + \frac{1}{64} + \frac{1}{125} + \frac{1}{216} \\
& + \frac{1}{343} + \frac{1}{512} + \frac{1}{729} + \frac{1}{1000} \\
& + \frac{1}{1331} + \frac{1}{1728} + \frac{1}{2197} \\
& + \frac{1}{2744} + \frac{1}{3375} + \frac{1}{4096} \\
& + \frac{1}{4913} + \frac{1}{5832} + \frac{1}{6859} \\
& + \frac{1}{8000} + \ldots \tag{2}
\end{autobreak}
\end{align}

The magic happens from the simple fact that \texttt{autobreak} interprets all new line characters appearing between \texttt{\begin{autobreak}} and \texttt{\end{autobreak}} as \textit{breakable points}, at which any line breaks can be logically inserted. To be more exact, the first non-empty block, separated from the rest by a new line character, determines the indentation of the successive lines. Then \texttt{autobreak} tries to fill the line with the rest of the blocks, and puts a line break when they do not fit in a line. This is clarified by the following example:

\footnote{Technically, \texttt{align} (with \texttt{\notag} to suppress equation numbers except the last line) is the only option we can use for page-break aligned equations within \texttt{amsmath} because \texttt{split}, \texttt{gathered}, \texttt{aligned} and \texttt{alignedat} do not allow page breaking. \texttt{dmath} of \texttt{breqn} with \texttt{\eqinterlinepenalty=0} allows page breaking, but may fail to find a reasonable tag place.}
It is also possible to put more than one `autobreak` in one `align`:

\begin{align}
\begin{autobreak}
\NumberedBox{1} = \NumberedBox{2} + \NumberedBox{3} + \NumberedBox{4} + \NumberedBox{5} + \NumberedBox{6} + \NumberedBox{7} + \NumberedBox{8} + \NumberedBox{9} + \NumberedBox{10} , \tag{3}
\end{autobreak}
\end{align}

\begin{align}
\begin{autobreak}
\LongerNumberedBox{1} = \NumberedBox{2} + \NumberedBox{3} + \NumberedBox{4} + \NumberedBox{5} + \NumberedBox{6} + \NumberedBox{7} + \NumberedBox{8} + \NumberedBox{9} + \NumberedBox{10} . \tag{4}
\end{autobreak}
\end{align}

\begin{align}
\begin{autobreak}
\LongerNumberedBox{1} = \NumberedBox{2} + \NumberedBox{3} + \NumberedBox{4} + \NumberedBox{5} + \NumberedBox{6} + \NumberedBox{7} + \NumberedBox{8} + \NumberedBox{9} + \NumberedBox{10} . \tag{5}
\end{autobreak}
```
\end{align}

For a technical reason, it often requires more than one run of \LaTeX, and in such
cases one will get informed by the following warning:

```
Package autobreak Warning: Layout may have changed.
    (autobreak) Rerun to get layout correct.
```

In the next run, the layout of the equations will be corrected.

\[ \MoveEqLeft \left( n_1 + n_2 + n_3 + n_4 \right)^3 = \]
\[ = \]
\[ + \quad n_1^3 \]
\[ + 3 \ n_1 \ n_2 \ n_2 \]
\[ + 3 \ n_1 \ n_2^2 \]
\[ + n_2^3 \]
\[ + 3 \ n_1^2 \ n_3 \]
\[ + 6 \ n_1 \ n_2 \ n_3 \]
\[ + 3 \ n_1 \ n_3^2 \]
\[ + 3 \ n_2 \ n_3^2 \]
\[ + n_3^3 \]
\[ + 3 \ n_1^2 \ n_4 \]
\[ + 6 \ n_1 \ n_2 \ n_4 \]
\[ + 3 \ n_1 \ n_3 \ n_4 \]
\[ + 3 \ n_2 \ n_3 \ n_4 \]
\[ + 3 \ n_3 \ n_4^2 \]
\[ + n_4^3 . \] (6)

This command is designed to work like `\MoveEqLeft` of the `mathtools` package\(^5\). If it is put at the beginning of an `autobreak` environment, then all subsequent lines after the first line are indented by 2 em (the default value).

\begin{align}
\begin{autobreak}
\MoveEqLeft \left( n_1 + n_2 + n_3 + n_4 \right)^3 = n_1^3 + n_2^3 + n_3^3 + n_4^3 =
\end{autobreak}
\end{align}

The indent width can be changed by an optional argument of the command.

\begin{verbatim}
\everybeforeautobreak {⟨tokens⟩}
\everyafterautobreak {⟨tokens⟩}  \end{verbatim}

\(^5\)https://www.ctan.org/pkg/mathtools.
They specify token lists inserted before and after automatically inserted line breaks in `autobreak`. For example,

\begin{align}
\begin{everyafterautobreak}\times\end{everyafterautobreak}
\begin{autobreak}
\cos\left(\frac{\pi x}{2}\right) &= \left(1-x^2\right)\left(1-\frac{x^2}{9}\right)\left(1-\frac{x^2}{25}\right)\left(1-\frac{x^2}{49}\right)\left(1-\frac{x^2}{81}\right)\left(1-\frac{x^2}{121}\right)\left(1-\frac{x^2}{169}\right)\left(1-\frac{x^2}{225}\right)\left(1-\frac{x^2}{289}\right)\left(1-\frac{x^2}{361}\right)\left(1-\frac{x^2}{441}\right)\ldots
\end{autobreak}
\end{align}

\section{Caveats}

Because `autobreak` tries to insert line breaks at any of new line characters, you must not make any new lines at which the line cannot be broken\(^6\). For example

\begin{align}
\begin{align}
\begin{autobreak}
\text{x =}
% A problematic line break.
\frac{1}{2}
\end{autobreak}
\end{align}
\end{align}

\(\text{You may put } \% \text{ at the end of the line to avoid a new line.}\)
gives an error in the typesetting:

```latex
! Missing } inserted.
<inserted text>
}
1.8 \end{align}
```

Putting `\` or `&` inside \texttt{autobreak}, which tries to insert these special stuffs automatically, also causes typesetting errors.

The \texttt{autobreak} environment uses \texttt{\linewidth} as the maximum width that expressions in its body can occupy. There is no way for \texttt{autobreak} to know how much other formulae consume the space outside it. Therefore it fails to determine the adequate maximum width when there are expressions outside \texttt{autobreak} and \LaTeX{} gives overfull line warnings:

```latex
\begin{align}
\begin{autobreak}
\text{some stuff outside autobreak}
\begin{autobreak}
\text{LHS} = \\
\text{...}
\end{autobreak} . \% Even just a "." can be problematic.
\end{autobreak}
\end{align}
```

May give overfull line warnings

One may want to separate long formulae from the main document file to other files and include them via `\input{⟨file⟩}` for example,

```latex
\begin{align}
\begin{autobreak}
\end{autobreak} % It works!
\end{align}
```

```latex
\begin{align}
\begin{autobreak}
\text{lhs} = \\
\text{...}
\end{autobreak} % It also works!
\end{align}
```

The current version of \texttt{autobreak} supports these cases: the file content of `\input{⟨file⟩}` is expanded before recognizing the lines, with the help of the \texttt{catchfile} package\footnote{https://www.ctan.org/pkg/catchfile}, when it appears at the beginning of each line. But it does not support `\input{⟨file⟩}` in the middle of the lines:

\footnote{https://www.ctan.org/pkg/catchfile.}
\begin{align}
\begin{autobreak}
x + \text{input{longexpr.inc}} \ % \text{Sorry, it does not work.}
\end{autobreak}
\end{align}

The difficulty comes from the fact that it needs to be expanded before \texttt{autobreak} scans lines. By the same reason, \texttt{autobreak} fails to detect new lines defined inside macros\textsuperscript{8}:

\begin{align}
\begin{autobreak}
\texttt{\newcommand{\foo}{a + b + c + d}}
\end{autobreak}
\end{align}

\begin{align}
\begin{autobreak}
\texttt{\foo + \foo + \foo + \foo \ % \text{No new lines can be detected.}}
\end{autobreak}
\end{align}

4 Implementation

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{autobreak} [2017/02/23 v0.3 simple line breaking of long formulae]
\end{verbatim}

4.1 Registers and constants

\begin{verbatim}
\everybeforeautobreak The list of tokens that gets inserted before every line break generated by \texttt{autobreak}.
\everyafterautobreak The list of tokens that gets inserted after every line break generated by \texttt{autobreak}.
@autobreak@alltoks The token register to store the whole result of \texttt{autobreak}.
@autobreak@linetoks The token register for the current line.
\end{verbatim}

\textsuperscript{8}Actually, when the definition of \texttt{\foo} is parsed, the new line characters inside it are usually lost.
4.2 Interaction with .aux files

When there are two or more `autobreak` in one `align`, each `autobreak` has to know the maximum width of the left-hand side of the all `autobreak` in the same `align`. Instead of violating ‘causality’ (e.g., how \LaTeX\ parses a file from the beginning to the end), we use .aux file to store the maximum left-hand side width, which provides the correct value in the next run.

Show a warning if the user needs to rerun.

\begin{Verbatim}
\if@autobreak@invalidlayout
  \if@filesw
    \PackageWarningNoLine{autobreak}{Layout may have changed.}
    \MessageBreak Rerun to get layout correct}
  \else
    \PackageWarningNoLine{autobreak}{Layout may be wrong}
  \fi
\fi
\end{Verbatim}
To be expanded to a value saved in .aux in the previous run, or 0pt if not found.

\@autobreak@getmaxlhswidth\def\@autobreak@getmaxlhswidth\#1{%\ifundefined{@autobreak@w@\#1}{\z@}{\@nameuse{@autobreak@w@\#1}}%
}

\@autobreak@setmaxlhswidth\def\@autobreak@setmaxlhswidth\#1\#2{%\global\@namedef{@autobreak@w@\#1}{\#2}%}

\@autobreak@eqnindex The counter to identify each align.
\newcounter{@autobreak@eqnindex}

\@autobreak@subeqnindex The counter to store the number of autobreak in an align.
\newcounter{@autobreak@subeqnindex}[@autobreak@eqnindex]

\@autobreak@loadmaxlhswidth\def\@autobreak@loadmaxlhswidth{%\stepcounter{@autobreak@eqnindex}\@autobreak@maxlhswidth=\@autobreak@getmaxlhswidth{\arabic{@autobreak@eqnindex}}\@autobreak@realmaxlhswidth=\z@
}

\@autobreak@savemaxlhswidth\def\@autobreak@savemaxlhswidth{%\ifnum\arabic{@autobreak@subeqnindex}>0\ifdim\@autobreak@maxlhswidth=\@autobreak@realmaxlhswidth\else\global\@autobreak@invalidlayouttrue\fi\fi\ifnum\arabic{@autobreak@subeqnindex}>1\if@filesw\@ifundefined{@autobreak@auxinited}{\immediate\write\@mainaux{\string\providecommand\string\@autobreak@setmaxlhswidth[2]{}%}}{\fi}}

We have used the wrong value of \@autobreak@maxlhswidth (was too much). Need to rerun.
\global\@autobreak@invalidlayouttrue
\fi

Note that \@autobreak@maxlhswidth becomes problematic only when two or more autobreak appear in one align. In the case with one autobreak, the default value 0pt is safe for the next run.
\ifnum\arabic{@autobreak@subeqnindex}>1\if@filesw\@ifundefined{@autobreak@auxinited}{\immediate\write\@mainaux{%\string\providecommand\string\@autobreak@setmaxlhswidth[2]{}%}}{\fi}}

We should provide \@autobreak@setmaxlhswidth in .aux.
4.3 Hacking amsmath

\RequirePackage{amsmath}

\if@autobreak@newlinedef
The switch to be turned on when \@autobreak@newlinedef applies.
\newif\if@autobreak@newlinedef
\@autobreak@newlinedeffalse
\@autobreak@newlinedef
Installs the definition of \textasciitilde M as a space. This is virtually harmless in math mode.
\begingroup
\catcode'\textasciitilde=\active
\gdef\@autobreak@newlinedef{\def\textasciitilde{ }}\@autobreak@newlinedeftrue
\endgroup
\collect@body
We need to override \collect@body such that it keeps \textasciitilde M.
\def\collect@body#1{\@envbody={\expandafter#1\expandafter{\the\@envbody}}\edef\process@envbody{\the\@envbody\noexpand\end{\@currenvir}}\@envbody=\@emptytoks\begin@stack{b}\begingroup
Actually, the following three lines need to be inserted to the original code.
\if@autobreak@newlinedef\catcode'\textasciitilde=\active\fi\expandafter\let\csname\@currenvir\endcsname=\collect@body\edef\process@envbody{\expandafter\noexpand\csname\@currenvir\endcsname\@emptytoks
\begin@stack{b}\begingroup
}
align  Hack align of amsmath.
\let\@autobreak@oldstart@align=\start@align
\def\start@align{%  
\@autobreak@loadmaxlhswidth  
\@autobreak@newline\def  
\@autobreak@oldstart@align  
}
\let\@autobreak@oldendalign=\endalign
\def\endalign{%  
\@autobreak@savemaxlhswidth  
\@autobreak@oldendalign  
}

4.4  autobreak environment

autobreak  Checks if we are in align (and \@autobreak@newline\def is applied), increments
the counter and collects its body via }\collect@body.
\newenvironment{autobreak}{%  
\if\@autobreak@newline\def  
\else  
\PackageError{autobreak}{%  
autobreak is not allowed here%  
}{%  
Use autobreak inside align.%  
}%  
\fi  
\stepcounter{\@autobreak@subeqnindex}  
\collect@body\@autobreak  
}{ }

\@autobreak  Called from }\collect@body. The parameter #1 is the whole body. It takes also #2
and #3, which are always }\end and autobreak, to remove them from the successive
tokens.
\def\@autobreak#1#2#3{%  
First, close the group of autobreak.  
\end{autobreak}%  
Then parse the given body of the environment and construct lines to be passed to
align.  
\@autobreak@init  
\def\@tempa{\expandafter\@autobreak@scanline  
\@autobreak@newline\char#1}  
\expandafter\@tempa\@autobreak@newline\char\@autobreak@end  
}

\@autobreak@init  Initialization.  
\def\@autobreak@init{%  
\@autobreak@alltoks={}%
Finalization. It generates the whole lines in one go.

```latex
\def\@autobreak@end{% 
\expandafter\@autobreak@addtoks\expandafter\@autobreak@alltoks
\expandafter{\the\@autobreak@linetoks}\the\@autobreak@alltoks
}
```

Takes a line from the input stream. Here a line ends with \texttt{^^M}.

```latex
\begingroup \catcode'\^^M=\active \gdef\@autobreak@scanline#1\^^M{\@autobreak@scanline@{#1}} \endgroup
```

If the next token is a punctuation, then we merge it into the current line. (Otherwise it can make a line only with a period, for example).

```latex
\def\@autobreak@scanline@#1{% 
\@autobreak@ifnextpunct{\@autobreak@scanline@gobble{#1}}{\@autobreak@scanline@@{#1}}%
}
```

A helper macro of \texttt{\@ifnextpunct{(if-yes)}{(if-no)}}.

```latex
\def\@autobreak@ifnextpunct#1#2{% 
\@ifnextchar.{#1}{\@ifnextchar,{#1}{\@ifnextchar;{#1}{#2}}}%
}
```

Merge punctuations as possible (usually there is only one period in a line, though).
Pass the current line to \@autobreak@processline. Then, repeat scanning lines until \@autobreak@end appears as the next token.

Catch MoveEqLeft.

The argument #1 is MoveEqLeft. This command accepts an optional number.

In the case with #1 = 0, a special treatment is required because \@autobreak@processline ignores a zero width. Insert a very tiny space.
In the case with $#1 < 0$, put a very tiny space, and then put the space with a positive width such that the first line is indented to the right.

One may expect \input{⟨file⟩} in \texttt{autobreak} is expanded by the file content and \texttt{autobreak} treats new lines in it correctly. But it needs more work. Because handling of \input{} in the middle of the lines is rather involved, for now we support only \input{} at the beginning of each line (which is what sane people usually do). This can be done via the \texttt{catchfile} package.

The argument $#1$ is \input{} and $#2$ is the file name.

\texttt{\@autobreak@MoveEqLeft} This definition is expanded only when \texttt{\@autobreak@scanline} cannot detect \texttt{\MoveEqLeft} in an \texttt{autobreak} environment, in other words, when it appears in the middle of a line.
Each line from `\autobreak@scanline` should be regarded as a ‘block’ in the equation. The first block (typically the left-hand side + ‘=’) determines the indentation for the successive lines. From the second block, try to append the block to the end of the line and insert a line break if it does not fit in a line. Note that we measure the widths of the blocks with putting `{}` around alignment tabs.

For the first block. The rest of the width for the right-hand sides is determined from `\linewidth` and `\@autobreak@maxlhswidth`.

The previous one used the wrong value of `\@autobreak@maxlhswidth` (was too short). Need to rerun.

For the rest of the blocks.

Adding the next block gives an overfull line. Need a line break.
\@autobreak@addtoks  Appends \#2 to the token register \#1.
280  \def\@autobreak@addtoks#1#2{%
281    #1=\expandafter{\the#1#2}%
282 }\@autobreak@settowidth
Same as \settowidth but in math mode. We assume \displaystyle. (Anyway
align issues \displaystyle at the beginning of every cell.)
283  \def\@autobreak@settowidth#1#2{%
284    \settowidth#1{$\displaystyle#2$}%
285 }\@autobreak

Change History

v0.1  General: Initial version ............. 1
v0.2  \@autobreak@processline: Fix alignment tabs ............. 16
      \@autobreak@scanline: Add space calculation around
      \MoveEqLeft command ...... 14

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
\@autobreak@linetoks .............
\@autobreak ................. 120, 122
\@autobreak@MoveEqLeft ....... 133, 243
\@autobreak@addtoks 136, 273, 277, 280
\@autobreak@alltoks .............
\@autobreak@loadmaxlhswidth . 42, 101
\@autobreak@maxlhswidth ...........
\@autobreak@maxrhswidth ...........
\@autobreak@newlinechar ...........
\@autobreak@newlinedef ...........
\@autobreak@newlinechar .............
\@autobreak@nextpunct 145, 151, 171
\@autobreak@init .............. 124, 129
\@autobreak@getmaxlhswidth ... 30, 45
\@autobreak@newlineedeffalse .... 74
\@autobreak@invalidlayoutfalse . 19
\@autobreak@invalidlayouttrue 52, 257
\@autobreak@realmaxlhswidth ....... 12, 46, 50, 66, 261, 262
<table>
<thead>
<tr>
<th>K</th>
<th>\kern .................................. 208, 209, 212, 213, 216, 217</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>\let .................................. 93, 99, 105, 133</td>
</tr>
<tr>
<td></td>
<td>\linewidth ...................................... 264</td>
</tr>
<tr>
<td>M</td>
<td>\MessageBreak .................................. 24, 233</td>
</tr>
<tr>
<td></td>
<td>\MoveEqLeft .................................. 133, 184, 245, 247</td>
</tr>
<tr>
<td>N</td>
<td>\NeedsTeXFormat ................................. 2</td>
</tr>
<tr>
<td></td>
<td>\newcounter .................................. 40, 41</td>
</tr>
<tr>
<td></td>
<td>\newdimen .................................. 9, 10, 11, 12, 13</td>
</tr>
<tr>
<td></td>
<td>\newenvironment .................................. 110</td>
</tr>
<tr>
<td></td>
<td>\newif .................................. 18, 73</td>
</tr>
<tr>
<td></td>
<td>\newtoks .................................. 5, 6, 7, 8</td>
</tr>
<tr>
<td></td>
<td>\noexpand .................................. 85, 95</td>
</tr>
<tr>
<td></td>
<td>\notag .................................. 274</td>
</tr>
<tr>
<td>P</td>
<td>\p@ .................................. 198, 200</td>
</tr>
<tr>
<td></td>
<td>\PackageError .................................. 113, 244</td>
</tr>
<tr>
<td></td>
<td>\PackageWarning .................................. 231</td>
</tr>
<tr>
<td></td>
<td>\z@ .................................. 32, 46, 132, 198, 200, 252, 254, 256</td>
</tr>
<tr>
<td>R</td>
<td>\RequirePackage .................................. 72, 220</td>
</tr>
<tr>
<td>S</td>
<td>\settowidth .................................. 284</td>
</tr>
<tr>
<td></td>
<td>\space .................................. 245, 247</td>
</tr>
<tr>
<td></td>
<td>\start@align .................................. 99, 100</td>
</tr>
<tr>
<td></td>
<td>\stepcounter .................................. 43, 119</td>
</tr>
<tr>
<td></td>
<td>\string .................................. 58, 59, 64</td>
</tr>
<tr>
<td>T</td>
<td>\the .................................. 66, 83, 85, 137, 138, 270, 272, 281</td>
</tr>
<tr>
<td>W</td>
<td>\write .................................. 57, 63</td>
</tr>
<tr>
<td>Z</td>
<td>\z@ .................................. 32, 46, 132, 198, 200, 252, 254, 256</td>
</tr>
</tbody>
</table>

19