The **fibnum** package

Heiko Oberdiek*

2016/05/16 v1.1

Abstract

The package **fibnum** provides expandable fibonacci numbers for both LaTeX and plain TeX.

Contents

1 Documentation

In the mailing list `texhax` Jan Abraham asked the question, how to get Fibonacci numbers in TeX [1]:

> Write a Macro in TeX that compute the function `\fib{m}` All fibonacci numbers from 1 to m ($m < 40$).

This packages provides an expandable implementation for the calculation of these numbers for a much larger set of indexes. For practical reasons the index is restricted to the same limitations that apply for TeX integer numbers. The range of the Fibonacci numbers, however, are not limited by the algorithm. They are only restricted to memory limitations, if they are hit.

The package is loaded as LaTeX package in LaTeX:

---

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

1

\usepackage{fibnum}

and as file in plain TeX:

        \input fibnum.sty

The package does not know any options and it provides the macros `\fibnum` and `\fibnumPreCalc`.

`\fibnum {⟨index⟩}`

Macro `\fibnum` expects a TeX number as ⟨index⟩ in the official TeX number range from $-(2^{31}-1)$ up to $2^{31}-1$. In exact two expansion steps the macro expands to the Fibnoacci number $F_{\langle index \rangle}$. In case of a negative ⟨index⟩, the "negafibonacci" number [2] is used. Formally the Fibonacci number F_n with integer index n, $n \in \mathbb{Z}$ and $n \in [-2\,147\,483\,647, 2\,147\,483\,647]$ that is returned by macro `\fibnum` with numerical argument n is defined the following way:

$$F_n = \begin{cases} 0 & \text{for } n = 0 \\ 1 & \text{for } n = 1 \\ F_{n-1} + F_{n-2} & \text{for } n > 1 \\ (-1)^{n+1} F_n & \text{for } n < 0 \end{cases} \tag{1}$$

Examples:

        \fibnum{-6}   → -8
        \fibnum{-5}   → 5
        \fibnum{-4}   → -3
        \fibnum{-3}   → 2
        \fibnum{-2}   → -1
        \fibnum{-1}   → 1
        \fibnum{0}   → 0
        \fibnum{1}   → 1
        \fibnum{2}   → 1
        \fibnum{3}   → 2
        \fibnum{4}   → 3
        \fibnum{5}   → 5
        \fibnum{6}   → 8
                ⋮
        \fibnum{10}   → 55
                ⋮
        \fibnum{46}   → 1836311903
                ⋮
        \fibnum{100} → 354224848179261915075
                ⋮
        \fibnum{200} → 280571172992510140037611932413038677189525
                ⋮
        \fibnum{1000} → 43466557686937456435688527675040625802564
                        60517371780402481729089536555417949051890
                        40387984007925516929592259308032263477520
                        96896232398733224711616429964409065331879
                        382989696499285160037044761377951668492288
                        75

`\fibnumPreCalc {⟨index⟩}`

The package already provides precalculated Fibonacci numbers up to index 46. That means that calculations are not necessary for Fibonacci numbers that fit

into the range of TeX numbers. Because macro `\fibnum` is expandable, it cannot store calculated Fibonacci numbers for later use. Macro definitions are forbidden in expandable contexts. If larger Fibonacci numbers are used more than once, than the compilation time can be shortened by calculating and storing the Fibonacci numbers beforehand. The argument $\langle index \rangle$ is a TeX number and macro `\fibnumPreCalc` ensures that the Fibonacci numbers F_0 up to $F_{|\langle index \rangle|}$ that are not already known are calculated and stored in internal macros. Internally only non-negative Fibonacci numbers are stored. If $\langle index \rangle$ is negative, then the needed positive Fibonacci numbers are calculated and stored. Example:

> `\fibnumPreCalc{50}`
> *% calculates and stores the values for indexes 47..50.*
> *% (Values for 0..46 are already stored by the package.)*
> `\fibnum{49}` *% uses the stored value*
> `\fibnum{51}` *% only calculates F_{51} from stored values F_{49} and F_{50}*
> `\fibnumPreCalc{100}`
> *% calculates and stores the values for indexes 51..100*
> `\fibnum{100}` *% uses the stored value for F_{100}*
> `\fibnum{-100}` *% uses the stored value for F_{100}*
> *% $F_{-100} = -F_{100}$ according to equation (1).*

2 Implementation

2.1 Identification

1 ⟨∗package⟩

Reload check, especially if the package is not used with LaTeX.

```
 2 \begingroup\catcode61\catcode48\catcode32=10\relax%
 3   \catcode13=5 % ^^M
 4   \endlinechar=13 %
 5   \catcode35=6 % #
 6   \catcode39=12 % '
 7   \catcode44=12 % ,
 8   \catcode45=12 % -
 9   \catcode46=12 % .
10   \catcode58=12 % :
11   \catcode64=11 % @
12   \catcode123=1 % {
13   \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@fibnum.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17   \def\empty{}%
18   \ifx\x\empty % LaTeX, first loading,
19     % variable is initialized, but \ProvidesPackage not yet seen
20   \else
21     \expandafter\ifx\csname PackageInfo\endcsname\relax
22       \def\x#1#2{%
23         \immediate\write-1{Package #1 Info: #2.}%
24       }%
25     \else
26       \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27     \fi
28     \x{fibnum}{The package is already loaded}%
29     \aftergroup\endinput
30   \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
```

```
34  \catcode13=5 % ^^M
35  \endlinechar=13 %
36  \catcode35=6 % #
37  \catcode39=12 % '
38  \catcode40=12 % (
39  \catcode41=12 % )
40  \catcode44=12 % ,
41  \catcode45=12 % -
42  \catcode46=12 % .
43  \catcode47=12 % /
44  \catcode58=12 % :
45  \catcode64=11 % @
46  \catcode91=12 % [
47  \catcode93=12 % ]
48  \catcode123=1 % {
49  \catcode125=2 % }
50  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51    \def\x#1#2#3[#4]{\endgroup
52      \immediate\write-1{Package: #3 #4}%
53      \xdef#1{#4}%
54    }%
55  \else
56    \def\x#1#2[#3]{\endgroup
57      #2[{#3}]%
58      \ifx#1\@undefined
59        \xdef#1{#3}%
60      \fi
61      \ifx#1\relax
62        \xdef#1{#3}%
63      \fi
64    }%
65  \fi
66  \expandafter\x\csname ver@fibnum.sty\endcsname
67  \ProvidesPackage{fibnum}%
68    [2016/05/16 v1.1 Fibonacci numbers (HO)]%
69  \begingroup\catcode61\catcode48\catcode32=10\relax%
70    \catcode13=5 % ^^M
71    \endlinechar=13 %
72    \catcode123=1 % {
73    \catcode125=2 % }
74    \catcode64=11 % @
75    \def\x{\endgroup
76      \expandafter\edef\csname FibNum@AtEnd\endcsname{%
77        \endlinechar=\the\endlinechar\relax
78        \catcode13=\the\catcode13\relax
79        \catcode32=\the\catcode32\relax
80        \catcode35=\the\catcode35\relax
81        \catcode61=\the\catcode61\relax
82        \catcode64=\the\catcode64\relax
83        \catcode123=\the\catcode123\relax
84        \catcode125=\the\catcode125\relax
85      }%
86    }%
87  \x\catcode61\catcode48\catcode32=10\relax%
88  \catcode13=5 % ^^M
89  \endlinechar=13 %
90  \catcode35=6 % #
91  \catcode64=11 % @
92  \catcode123=1 % {
93  \catcode125=2 % }
94  \def\TMP@EnsureCode#1#2{%
95    \edef\FibNum@AtEnd{%
```

```
96     \FibNum@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{33}{12}% !
102 %\TMP@EnsureCode{36}{3}% $
103 %\TMP@EnsureCode{38}{4}% &
104 \TMP@EnsureCode{40}{12}% (
105 \TMP@EnsureCode{41}{12}% )
106 \TMP@EnsureCode{45}{12}% -
107 \TMP@EnsureCode{46}{12}% .
108 \TMP@EnsureCode{47}{12}% /
109 \TMP@EnsureCode{58}{12}% :
110 \TMP@EnsureCode{60}{12}% <
111 \TMP@EnsureCode{62}{12}% >
112 \TMP@EnsureCode{91}{12}% [
113 %\TMP@EnsureCode{96}{12}% `
114 \TMP@EnsureCode{93}{12}% ]
115 %\TMP@EnsureCode{94}{12}% ^ (superscript) (!)
116 %\TMP@EnsureCode{124}{12}% |
117 \edef\FibNum@AtEnd{\FibNum@AtEnd\noexpand\endinput}
```

2.2   Package resources

```
118 \begingroup\expandafter\expandafter\expandafter\endgroup
119 \expandafter\ifx\csname RequirePackage\endcsname\relax
120   \def\TMP@RequirePackage#1[#2]{%
121     \begingroup\expandafter\expandafter\expandafter\endgroup
122     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
123       \input #1.sty\relax
124     \fi
125   }%
126   \TMP@RequirePackage{ltxcmds}[2011/04/18]%
127   \TMP@RequirePackage{intcalc}[2007/09/27]%
128   \TMP@RequirePackage{bigintcalc}[2007/11/11]%
129 \else
130   \RequirePackage{ltxcmds}[2011/04/18]%
131   \RequirePackage{intcalc}[2007/09/27]%
132   \RequirePackage{bigintcalc}[2007/11/11]%
133 \fi
```

2.3   Setup precalculated values

```
134 \def\FibNum@temp#1{%
135   \expandafter\def\csname FibNum@#1\endcsname
136 }
137 \catcode46=9 % dots are ignored
138 \FibNum@temp{0}{0}
139 \FibNum@temp{1}{1}
140 \FibNum@temp{2}{1}
141 \FibNum@temp{3}{2}
142 \FibNum@temp{4}{3}
143 \FibNum@temp{5}{5}
144 \FibNum@temp{6}{8}
145 \FibNum@temp{7}{13}
146 \FibNum@temp{8}{21}
147 \FibNum@temp{9}{34}
148 \FibNum@temp{10}{55}
149 \FibNum@temp{11}{89}
150 \FibNum@temp{12}{144}
151 \FibNum@temp{13}{233}
152 \FibNum@temp{14}{377}
153 \FibNum@temp{15}{610}
```

```
154 \FibNum@temp{16}{987}
155 \FibNum@temp{17}{1.597}
156 \FibNum@temp{18}{2.584}
157 \FibNum@temp{19}{4.181}
158 \FibNum@temp{20}{6.765}
159 \FibNum@temp{21}{10.946}
160 \FibNum@temp{22}{17.711}
161 \FibNum@temp{23}{28.657}
162 \FibNum@temp{24}{46.368}
163 \FibNum@temp{25}{75.025}
164 \FibNum@temp{26}{121.393}
165 \FibNum@temp{27}{196.418}
166 \FibNum@temp{28}{317.811}
167 \FibNum@temp{29}{514.229}
168 \FibNum@temp{30}{832.040}
169 \FibNum@temp{31}{1.346.269}
170 \FibNum@temp{32}{2.178.309}
171 \FibNum@temp{33}{3.524.578}
172 \FibNum@temp{34}{5.702.887}
173 \FibNum@temp{35}{9.227.465}
174 \FibNum@temp{36}{14.930.352}
175 \FibNum@temp{37}{24.157.817}
176 \FibNum@temp{38}{39.088.169}
177 \FibNum@temp{39}{63.245.986}
178 \FibNum@temp{40}{102.334.155}
179 \FibNum@temp{41}{165.580.141}
180 \FibNum@temp{42}{267.914.296}
181 \FibNum@temp{43}{433.494.437}
182 \FibNum@temp{44}{701.408.733}
183 \FibNum@temp{45}{1.134.903.170}
184 \FibNum@temp{46}{1.836.311.903}
```

\FibNum@max

```
185 \def\FibNum@max{46}
```

2.4   Macros for precalculating values

\fibnumPreCalc

```
186 \def\fibnumPreCalc#1{%
187   \expandafter\expandafter\expandafter
188   \FibNum@PreCalc\intcalcNum{#1}/%
189 }
```

\FibNum@PreCalc

```
190 \def\FibNum@PreCalc#1/{%
191   \ifnum#1<\ltx@zero
192     \expandafter\FibNum@PreCalc\ltx@gobble#1/%
193   \else
194     \ifnum#1>\FibNum@max
195       \begingroup
196         \ltx@LocDimenA=#1sp\relax
197         \countdef\FibNum@i=255\relax
198         \FibNum@i=\FibNum@max\relax
199         \edef\FibNum@temp{%
200           \csname FibNum@\the\FibNum@i\endcsname/%
201         }%
202         \advance\FibNum@i by -1\relax
203         \edef\FibNum@temp{%
204           \FibNum@temp
205           \csname FibNum@\the\FibNum@i\endcsname
206         }%
207         \advance\FibNum@i\ltx@two
```

```
208        \iftrue
209          \expandafter\FibNum@PreCalcAux\FibNum@temp
210        \fi
211      \endgroup
212    \fi
213  \fi
214 }
```

\FibNum@PreCalcAux

```
215 \def\FibNum@PreCalcAux#1/#2\fi{%
216  \fi
217  \edef\FibNum@temp{\BigIntCalcAdd#1!#2!}%
218  \global\expandafter
219  \let\csname FibNum@\the\FibNum@i\endcsname\FibNum@temp
220  \ifnum\FibNum@i=\ltx@LocDimenA
221    \xdef\FibNum@max{\the\FibNum@i}%
222  \else
223    \advance\FibNum@i\ltx@one
224    \expandafter\FibNum@PreCalcAux\FibNum@temp/#1%
225  \fi
226 }
```

2.5   Expandable calculations

\fibnum

```
227 \def\fibnum#1{%
228  \romannumeral
229  \expandafter\expandafter\expandafter\FibNum@Do\intcalcNum{#1}/%
230 }
```

\FibNum@Do

```
231 \def\FibNum@Do#1/{%
232  \ifnum#1<\ltx@zero
233    \FibNum@ReturnAfterElseFiFi{%
234      \ifodd#1 %
235        \expandafter\expandafter\expandafter\ltx@zero
236      \else
237        \expandafter\expandafter\expandafter\ltx@zero
238        \expandafter\expandafter\expandafter-%
239      \fi
240      \romannumeral
241      \expandafter\FibNum@Do\ltx@gobble#1/%
242    }%
243  \else
244    \ifnum\FibNum@max<#1 %
245      \ltx@ReturnAfterElseFi{%
246        \expandafter
247        \FibNum@ExpCalc\number\expandafter\IntCalcInc\FibNum@max!%
248        \expandafter\expandafter\expandafter/%
249        \csname FibNum@\FibNum@max
250        \expandafter\expandafter\expandafter\endcsname
251        \expandafter\expandafter\expandafter/%
252        \csname FibNum@\expandafter\IntCalcDec\FibNum@max!%
253        \endcsname/%
254        #1%
255      }%
256    \else
257      \expandafter\expandafter\expandafter\ltx@zero
258      \csname FibNum@#1\expandafter\expandafter\expandafter\endcsname
259    \fi
260  \fi
261 }
```

`\FibNum@ReturnAfterElseFiFi`

```
262 \def\FibNum@ReturnAfterElseFiFi#1\else#2\fi\fi{\fi#1}
```

`\FibNum@ExpCalc`

```
263 \def\FibNum@ExpCalc#1/#2/#3/#4\fi{%
264   \fi
265   \ifnum#1=#4 %
266     \ltx@ReturnAfterElseFi{%
267       \expandafter\expandafter\expandafter\ltx@zero
268       \BigIntCalcAdd#2!#3!%
269     }%
270   \else
271     \expandafter\FibNum@ExpCalc
272     \number\IntCalcInc#1!%
273     \expandafter\expandafter\expandafter/%
274     \BigIntCalcAdd#2!#3!/%
275     #2/#4%
276   \fi
277 }
278 \FibNum@AtEnd%
279 ⟨/package⟩
```

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/fibnum.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/fibnum.pdf Documentation.

Bundle. All the packages of the bundle 'oberdiek' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TeX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain TeX:

```
tex fibnum.dtx
```

---

[1] CTAN:pkg/fibnum

8

TDS.   Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
fibnum.sty → tex/generic/oberdiek/fibnum.sty
fibnum.pdf → doc/latex/oberdiek/fibnum.pdf
fibnum.dtx → source/latex/oberdiek/fibnum.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

3.4   Refresh file name databases

If your TeX distribution (TeX Live, MiKTeX, . . .) relies on file name databases, you must refresh these. For example, TeX Live users run `texhash` or `mktexlsr`.

3.5   Some details for the interested

Unpacking with LaTeX.   The `.dtx` chooses its action depending on the format:

plain TeX: Run docstrip and extract the files.

LaTeX: Generate the documentation.

If you insist on using LaTeX for docstrip (really, docstrip does not need LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{fibnum.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation.   You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLaTeX:

```
pdflatex fibnum.dtx
bibtex fibnum.aux
makeindex -s gind.ist fibnum.idx
pdflatex fibnum.dtx
makeindex -s gind.ist fibnum.idx
pdflatex fibnum.dtx
```

4   References

[1] Jan Abraham. *[texhax] Beginner in TEX MACRO to compute functions.* 2012-04-07. URL: https://tug.org/pipermail/texhax/2012-April/019146.html (visited on 2012-04-08).

[2] Wikipedia contributors. *Fibonacci numbers.* Version 486266088. Wikipedia, The Free Encyclopedia. 2012-04-08. URL: https://en.wikipedia.org/w/index.php?title=Fibonacci_number&oldid=486266088 (visited on 2012-04-08).

5   History

[2012/04/08 v1.0]

- First version.

[2016/05/16 v1.1]

- Documentation updates.

6 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; plain numbers refer to the code lines where the entry is used.