Athenian Numerals

Apostolos Syropoulos
366, 28th October Str.
GR-671 00 Xanthi, HELLAS
Email:apostolo@platon.ee.duth.gr

1997/09/19

1 Introduction

`\athnum` This little LaTeX package implements the macro `\athnum`. The macro transforms an Arabic numeral, i.e., the kind of numerals we all use (e.g., 1, 5, 789 etc), to the corresponding *Athenian* numeral. Athenian numerals were in use only in ancient
`\PiIt` Athens. The special multiples, which the system employs, are drawn with the special macro `\PiIt`. The macro produces a more or less Π-like shape above a letter.

2 The Numbering System

The athenian numbering system, like the roman one, employs letters to denote important numbers. Multiple occurrence of a letter denote a multiple of the "important" number, e.g., the letter I denotes 1, so III denotes 3. Here are the basic digits used in the Athenian numbering system:

- I denotes the number one (1)
- Π denotes the number five (5)
- Δ denotes the number ten (10)
- H denotes the number one hundred (100)
- X denotes the number one thousand (1000)
- M denotes the number ten thousands (10000)

Moreover, the letters Δ, H, X, and M under the letter Π, denote five times their original value, e.g., the symbol $\boxed{\text{X}}$, denotes the number 5000, and the symbol $\boxed{\Delta}$, denotes the number 50. It must be noted that the numbering system does not provide negative numerals or a symbol for zero.

The Athenian numbering system is described, among others, in an article in Encyclopedia Δ*ομή*, Vol. 2, page 280, 7th edition, Athens, October 2, 1975.

3 The Code

Before we do anything further, we have to identify the package.

```
⟨*package⟩
\ProvidesPackage{grnumalt}[1997/09/19\space v1.1]
\typeout{Package: ‘grnumalt’ v1.0\space <1997/09/19> (AS)}
```

\PiIt It is very important to be able to correctly typeset the multiples of the numbering system. For this purpose we define the macro `\PiIt`. The macro uses two “length” variables.

```
\newdimen\@boxW \newdimen\@boxH
```

We make the `\PiIt` macro a robust command.

```
\DeclareRobustCommand{\PiIt}[1]{%
```

In order to correctly produce the Π symbol we need to know the height and width of the letter that goes under a Π. This is done by using the standard LaTeX macros: `\settowidth` and `\settoheight`.

```
    \settowidth{\@boxW}{#1}%
    \settoheight{\@boxH}{#1}%
```

Since, the width of an ordinary rule is 0.4 pt we must add 0.8 pt to the width of the letter.

```
    \addtolength{\@boxW}{0.8pt}
```

Now comes the interesting part: the actual drawing. We create a vertical box. Inside this box we draw a horizontal rule of width equal to the width of the letter. Next, we create a horizontal box in order to make the vertical lines. We draw the first vertical line, then we put the letter in a `\mbox`, since it may be a mathematical symbol[1]. After the `\mbox` we draw the second vertical line and we “close” the horizontal box. A little white space is put after the vertical box, so that adjacent multiplies do not look ugly!

```
    \vbox{%
    \hrule width\@boxW\hbox{%
          \vrule height\@boxH\mbox{#1}%
          \vrule height\@boxH}}\kern.5pt}
```

\athnum Now, we turn our attention to the definition of the macro `\athnum`. This macro uses one integer variable.

```
\newcount\@ath@num
```

The macro `\athnum` is also defined as a robust command.

```
\DeclareRobustCommand{\athnum}[1]{%
```

The macro does not work in math mode so we must ensure that it will not be used in math mode. We could use `\ensuremath`, but our definition is too long...

```
       \ifmmode
         \errhelp{^^J This macro has been defined to work^^J
                   *only* in non-math mode. It is definitely^^J
                   sure that you are  using it in math mode.^^J}%
         \errmessage{^^JYou can’t use macro atheniannumeral^^J
                       in math mode.^^J}%
```

[1] greek letters are considered mathematical symbols by TeX.

If we are not in math mode, we can start computing the Athenian numeral. After assigning to variable `\@ath@num` the value of the macro's argument, we make sure that the argument is in the expected range, i.e., it is greater than zero. In case it isn't we simply produce a `\space`, warn the user about it and quit.

```
\else\@ath@num#1\relax
\ifnum\@ath@num=\z@%
  \space%
  \PackageWarning{grnumalt}{%
   Illegal value (\the\@ath@num) for athenian numeral}%
\else\ifnum\@ath@num<\z@%
  \space%
  \PackageWarning{grnumalt}{%
  Illegal value (\the\@ath@num) for athenian numeral}%
\else$
```

Having done all the necessary checks, we are now ready to do the actual computation. If the number is greater than 49999, then it certainly has at least one 𐅥 "digit". We find all such digits by continuously subtracting 50000 from `\NumA`, until `\NumA` becomes less than 50000.

```
\loop\ifnum\@ath@num>49999
   \PiIt{$\mathrm{M}$}
   \advance\@ath@num-50000
\repeat
```

We now check for tens of thousands.

```
\loop\ifnum\@ath@num>9999
\mathrm{M}\advance\@ath@num-\@M
\repeat
```

Since a number can have only on 𐅦 "digit" (equivalent to 5000), it is easy to check it out and produce the corresponding numeral in case it does have one.

```
\ifnum\@ath@num>4999
   \PiIt{$\mathrm{X}$}
   \advance\@ath@num-5000
\fi
```

Next, we check for thousands, the same way we checked for tens of thousands.

```
\loop\ifnum\@ath@num>999
\mathrm{X}\advance\@ath@num-\@m
\repeat
```

Like the five thousands, a numeral can have at most one 𐅧 "digit" (equivalent to 500).

```
\ifnum\@ath@num>499
   \PiIt{$\mathrm{H}$}
   \advance\@ath@num-500
\fi
```

It is time to check hundreds, which follow the same pattern as thousands

```
\loop\ifnum\@ath@num>99
\mathrm{H}\advance\@ath@num-100
\repeat
```

A numeral can have only one 𐅨 "digit" (equivalent to 50).

```
\ifnum\@ath@num>49
   \PiIt{$\Delta$}
```

```
54              \advance\@ath@num-50
55           \fi
```

Let's check now decades.

```
56           \loop\ifnum\@ath@num>9
57           \Delta\advance\@ath@num by-10
58           \repeat
```

We check for fives and, finally, for the digits 1, 2, 3, and 4.

```
59           \ifnum\@ath@num>4
60              \Pi
61              \advance\@ath@num-5
62           \fi
63           \ifcase\@ath@num
64              \or\mathrm{I}
65              \or\mathrm{II}
66              \or\mathrm{III}
67              \or\mathrm{IIII}
68           \fi$
69   \fi\fi\fi}
```

`\@athnum` The command `\@athnum` is defined just to make it possible to have, e.g., page numbering with athenian numerals.

```
70 \let\@athnum\athnum
71 ⟨/package⟩
```

Dedication

I would like to dedicate this piece of work to my son Demetrios-Georgios.