# The Polynom Package


Copyright 2000–2002
Carsten Heinz < `cheinz@gmx.de` >


2002/01/10 Version 0.14


**Abstract**

The polynom package implements macros for manipulating polynomials, for example it can typeset long polynomial divisions. Such long divisions can be shown step by step. The main test case and application is the polynomial ring in one variable with rational coefficients.


## 1 Introduction

Donald Arseneau has contributed a lot of packages to the TeX community. In particular, he posted macros for long division on `comp.text.tex`, which were also published in the TUGboat [1]. With these definitions, one could just write `\longdiv{12345}{13}` to get the long division shown in Figure 1 (a). In fact, that

$$
\begin{array}{r@{\,}r}
 & 949 \\
\cline{2-2}
13\,\big) & 12345 \\
 & 11700 \\
\cline{2-2}
 & 645 \\
 & 520 \\
\cline{2-2}
 & 125 \\
 & 117 \\
\cline{2-2}
 & 8
\end{array}
$$

(a) `\longdiv{12345}{13}`

$$
\begin{array}{r@{\,}rrrr}
 & & & & X^2 + 2X + 2 \\
\cline{2-5}
X - 1\,\big) & X^3 & + X^2 & & -1 \\
 & - X^3 & + X^2 & & \\
\cline{2-3}
 & & 2X^2 & & \\
 & & - 2X^2 & + 2X & \\
\cline{3-4}
 & & & 2X & -1 \\
 & & & - 2X & + 2 \\
\cline{4-5}
 & & & & 1
\end{array}
$$

(b) `\polylongdiv{X^3+X^2-1}{X-1}`

Figure 1: Integer and polynomial long division

integer long division has been typeset using the code from the location cited. The polynom package allows to do the similar job with polynomials, see Figure 1 (b). Figure 2 shows partial long divisions.

An application of polynomial division is shown in Figure 3: the Euclidean algorithm to determine a greatest common divisor of two polynomials. Note that in the case here, a greatest common divisor is uniquely determined up to a scalar factor, so $X - 1$ and $\frac{4}{9}X - \frac{4}{9}$ are both greatest common divisors in the example.

```
\polylongdiv[stage=1]
            {(X-1)(X^2+2X+2)+1}{X-1}
```

$$\begin{array}{r|rrrr} \multicolumn{1}{r}{} & & & & \\ \cline{2-5} X-1 & X^3 & +X^2 & & -1 \end{array}$$

```
\polylongdiv[stage=2]
            {(X-1)(X^2+2X+2)+1}{X-1}
```

$$\begin{array}{r|rrrr} \multicolumn{1}{r}{} & & X^2 & & \\ \cline{2-5} X-1 & X^3 & +X^2 & & -1 \end{array}$$

```
\polylongdiv[stage=3]
            {(X-1)(X^2+2X+2)+1}{X-1}
```

$$\begin{array}{r|rrrr} \multicolumn{1}{r}{} & & X^2 & & \\ \cline{2-5} X-1 & X^3 & +X^2 & & -1 \\ & -X^3 & +X^2 & & \\ \cline{2-3} \end{array}$$

```
\polylongdiv[stage=4]
            {(X-1)(X^2+2X+2)+1}{X-1}
```

$$\begin{array}{r|rrrr} \multicolumn{1}{r}{} & & X^2 & & \\ \cline{2-5} X-1 & X^3 & +X^2 & & -1 \\ & -X^3 & +X^2 & & \\ \cline{2-3} & & 2X^2 & & \end{array}$$

$\vdots$

Figure 2: Stepwise polynomial long division. The whole division is shown with `stage=10`. Note that other printing styles, see table 7, might require one more stage to put the remainder next to the result

$$\begin{aligned} X^4 - 2X^3 + 2X^2 - 2X + 1 &= \left(X^3 + X^2 - X - 1\right) \cdot \left(X - 3\right) + \left(6X^2 - 4X - 2\right) \\ X^3 + X^2 - X - 1 &= \left(6X^2 - 4X - 2\right) \cdot \left(\tfrac{1}{6}X + \tfrac{5}{18}\right) + \left(\tfrac{4}{9}X - \tfrac{4}{9}\right) \\ 6X^2 - 4X - 2 &= \left(\tfrac{4}{9}X - \tfrac{4}{9}\right) \cdot \left(\tfrac{27}{2}X + \tfrac{9}{2}\right) + 0 \end{aligned}$$

```
\polylonggcd {(X-1)(X-1)(X^2+1)} {(X-1)(X+1)(X+1)}
```

Figure 3: Euclidean algorithm with polynomials; the last nonzero remainder is a greatest common divisor

```
\polyfactorize {(X-1)(X-1)(X^2+1)}
```

$$\left(X^2 + 1\right)\left(X - 1\right)^2$$

```
\polyfactorize {2X^3+X^2-7X+3}
```

$$2\left(X - \tfrac{1}{2}\right)\left(X + \tfrac{1}{2} + \tfrac{\sqrt{13}}{2}\right)\left(X + \tfrac{1}{2} - \tfrac{\sqrt{13}}{2}\right)$$

```
\polyfactorize {120X^5-274X^4+225X^3-85X^2+15X-1}
```

$$120\left(X - 1\right)\left(X - \tfrac{1}{2}\right)\left(X - \tfrac{1}{3}\right)\left(X - \tfrac{1}{4}\right)\left(X - \tfrac{1}{5}\right)$$

Figure 4: Factorizations of some polynomials

## 2 Commands

The tables 5 and 6 list the user commands defined by this package. Each $\langle cs_{\dots}\rangle$ stands either for a TEX control sequence to store the internal representation of the result, or it stands for a previously saved result to operate with. The polynomials $\langle a\rangle$ and $\langle b\rangle$ are 'verbatim' polynomials as you would type them in math mode:[1] you may use `+`, `-`, `*`, `\cdot`, `/`, `\frac`, `(`, `)`, natural numbers, symbols like `e`, `\pi`, `\chi`, `\lambda`, and variables; the power operator `^` with integer exponents can be used on symbols, variables, and parenthesized expressions. Never use variables in a nominator, denominator or divisor.

| | |
|---|---|
| print long division $a/b$ (maybe partially) | `\polylongdiv[`$\langle keys\rangle$`]{`$\langle a\rangle$`}{`$\langle b\rangle$`}`<br>`\polylongdiv[`$\langle keys\rangle$`]`$\langle cs_a\rangle\langle cs_b\rangle$ |
| print Euclidean algorithm for $\gcd(a,b)$ | `\polylonggcd{`$\langle a\rangle$`}{`$\langle b\rangle$`}`<br>`\polylonggcd`$\langle cs_a\rangle\langle cs_b\rangle$ |
| print factorization of the polynomial $a$ | `\polyfactorize{`$\langle a\rangle$`}`<br>`\polyfactorize`$\langle cs_a\rangle$ |

Table 5: High-level user commands. The optional argument of the `\polylongdiv` command changes settings for the particular division

| | |
|---|---|
| $\langle cs_{a+b}\rangle \leftarrow a+b$ | `\polyadd`$\langle cs_{a+b}\rangle$`{`$\langle a\rangle$`}{`$\langle b\rangle$`}`<br>`\polyadd`$\langle cs_{a+b}\rangle\langle cs_a\rangle\langle cs_b\rangle$ |
| $\langle cs_{a-b}\rangle \leftarrow a-b$ | `\polysub`$\langle cs_{a-b}\rangle$`{`$\langle a\rangle$`}{`$\langle b\rangle$`}`<br>`\polysub`$\langle cs_{a-b}\rangle\langle cs_a\rangle\langle cs_b\rangle$ |
| $\langle cs_{ab}\rangle \leftarrow a\cdot b$ | `\polymul`$\langle cs_{ab}\rangle$`{`$\langle a\rangle$`}{`$\langle b\rangle$`}`<br>`\polymul`$\langle cs_{ab}\rangle\langle cs_a\rangle\langle cs_b\rangle$ |
| $\langle cs_{a/b}\rangle \leftarrow \lfloor a/b\rfloor$<br>`\polyremainder` $\leftarrow a \bmod b$ | `\polydiv`$\langle cs_{a/b}\rangle$`{`$\langle a\rangle$`}{`$\langle b\rangle$`}`<br>`\polydiv`$\langle cs_{a/b}\rangle\langle cs_a\rangle\langle cs_b\rangle$ |
| $\langle cs_{\mathrm{gcd}}\rangle \leftarrow \gcd(a,b)$ | `\polygcd`$\langle cs_{\mathrm{gcd}}\rangle$`{`$\langle a\rangle$`}{`$\langle b\rangle$`}`<br>`\polygcd`$\langle cs_{\mathrm{gcd}}\rangle\langle cs_a\rangle\langle cs_b\rangle$ |
| print polynomial $a$ | `\polyprint{`$\langle a\rangle$`}`<br>`\polyprint`$\langle cs_a\rangle$ |

Table 6: Low-level user commands

Note, however, that the support of symbols is very limited and that there is neither support of functions like $\sin(x)$ or $\exp(x)$, nor of roots or exponents other than integers, for example $\sqrt{\pi}$ or $e^x$. For teaching purposes this shouldn't be a major drawback. Particularly because there is a simple workaround in some cases:

[1]The scanner is based on the scanner of the `calc` package [2]. Read its documentation and the implementation part here if you want to know more.

the package doesn't look at symbols closely, so define the function or 'composed symbol' like $\sqrt{\pi}$ as a symbol. For example, you could write

```
\newcommand\epowerx{e^x}
\[\polylongdiv{\epowerx x^3-\epowerx x^2+\epowerx x-\epowerx}{x-1}\]
```

$$
\begin{array}{l}
e^x x^3 - e^x x^2 + e^x x - e^x = \left(x-1\right)\left(e^x x^2 + e^x\right) \\
\underline{- e^x x^3 + e^x x^2} \\
\qquad\qquad e^x x - e^x \\
\qquad\quad \underline{- e^x x + e^x} \\
\qquad\qquad\qquad 0
\end{array}
$$

Here the quotient and remainder are written next to dividend and divisor. This '`style=B`-feature' is discussed in the next section.

Let's conclude this section with an example of the low-level commands. If we want to divide $(X^2+X+1)(X-1)-\frac{\pi}{2}$ by $X-1$ and print quotient and remainder, we could do it like this:

```
\polydiv\polya {(X^2+X+1)(X-1)-\frac\pi2} {X-1}
`The quotient is \polyprint\polya,
the remainder \polyprint\polyremainder.'
```

Of course, we all know the result, so it isn't shown here. The calculation alone could also be done by, for example,

```
\polymul\polya {X^2+X+1} {X-1}
\polysub\polya \polya {\frac{\pi}{2}}
\polydiv\polya \polya {X-1}
```

# 3 Keys

`\polyset` Predefined variables are `X` and `x`, the default style for printing long division is shown in Figure 1 (b), and the left and right delimiters are `\left(` and `\right)`. The macro `\polyset` changes such default settings and accepts a comma separated list of 'key=value' pairs. This is implemented using the keyval package [3]. Possible keys and values are given in table 7. To make the style selection `C` clear and to use variables and delimiters other than the default, look at the output of the following example.

```
\begin{center}
\polyset{vars=XYZ\xi,  % make X, Y, Z, and \xi into variables
         style=C,      % use nonstandard style
         delims={[}{]}}% nongrowing brackets
\polylongdiv{(\xi-1)(\xi^2+2\xi+2)+1}{\xi-1}
\end{center}
```

$$
\begin{array}{l}
[\ \ \xi^3 \ + \xi^2 \qquad\quad - 2] \div [\xi - 1] = \xi^2 + 2\xi + 2 \\
\underline{\ - \xi^3 \ + \xi^2} \\
\qquad\quad 2\xi^2 \\
\qquad \underline{- 2\xi^2 + 2\xi} \\
\qquad\qquad\quad 2\xi - 2 \\
\qquad\qquad \underline{- 2\xi + 2} \\
\qquad\qquad\qquad\quad 0
\end{array}
$$

Afterwards previous settings are restored since the changes are made inside the center environment or, more generally, inside a group. The same `\polylongdiv` command now makes no sense. A constant would be divided by a constant.

| | |
|---|---|
| `vars=`⟨*token string*⟩ | make each token a variable |
| `style=`⟨`A`\|`B`\|`C`⟩ | define style for long division |
| `div=`⟨*token*⟩ | define division sign for `style=C`, default is ÷ |
| `stage=`⟨*number*⟩ | print long division up to stage ⟨*number*⟩ (starting with 1) |
| `delims={`⟨*left*⟩`}{`⟨*right*⟩`}` | define delimiters used for parenthesized expressions |

Table 7: Keys and values. The `delims` key has in fact an optional argument for use with growing delimiters. In this case you must specify invisible versions of the two delimiters. The default is `delims=[{\left.}{\right.}]{\left(}{\right)}`

## References


[1] Barbara Beeton and Donald Arseneau.
*Long division.*
In Jeremy Gibbons' *Hey — it works!*, TUGboat 18(2), June 1997, p. 75.

[2] Kresten Krab Thorup, Frank Jensen, and Chris Rowley.
*The* `calc` *package, Infix notation arithmetic in LaTeX*, 1998/07/07.
Available from `CTAN: macros/latex/required/tools`.

[3] David Carlisle.
*The* keyval *package*, 1999/03/16.
Available from `CTAN: macros/latex/required/graphics`.