0.1 General Information

The files *math.sty* and *preamble.sty* should provide you a simple yet effectie suite of macros for quick writing of mathematical/scientific papers. To properly load them you should include the following in your preamble:

```
\usepackage{preamble}
\usepackage{math}
```

It is important that you maintain the order of the packags, since math.sty uses some packages included in preamble.sty. Other than providing an extensive list of mathematical operators from math, there are some useful commands in preamble.sty too. The one that I myself use quite often is `\col{<color>}{<text>}`. Altough xcolor defines `\textcolor`, it can get kind of "clunky" in tables or similar, so i wrote a shorter command.

0.2 Symbol Index

Symbol	**Math-Mode**	**Result**	**Symbol**	**Math-Mode**	**Result**
Symbol			**Math-Mode**	**Result**	
Vectors					
Column Vector	`\pvec{x_1}{x_2}`	$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$	Dot-Product	`\dotp{x_1}{x_2}`	$\langle x_1, x_2 \rangle$
Column Vector	`\tvec{x_1}{x_2}{x_3}`	$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$			
Row Vector	`\rpvec{x_1}{x_2}`	$\begin{bmatrix} x_1 & x_2 \end{bmatrix}$			
Row Vector	`\rtvec{x_1}{x_2}{x_3}`	$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$			
Matrices					
Matrix	`\mat{M}`	$\mathbf{M}$	Matrix (greek letters)	`\gmat{M}`	$\boldsymbol{\phi}$
Determinant	`\det`	det			
Row Operation (arrow)	`\longleadsto{0.7}`	$\rightsquigarrow$	Row Operation	`\rtrf{r_2: -2r_1}`	$\overset{r_2:-2r_1}{\rightsquigarrow}$
Matrix-Rank (de)	`\Rang`	Rang	Matrix-Rank (en)	`\Rank`	Rank
Matrix-Trace (de)	`\Spur`	Spur	Matrix-Trace (en)	`\Trace`	Trace
Adjunct-Matrix	`\Adj`	Adj	Cofactor-Matrix	`\Cof`	Cof
Identity-Matrix (de)	`\imate`	$\mathbf{E}$	Identity-Matrix (en)	`\imati`	$\mathbf{I}$

Calculus and Functions					
Differential d	`\diff`	d	Exterior Derivative	`\extd`	D
Divergence	`\divs`	div			
Derivative	`\der{f}{x}`	$\frac{\mathrm{d}f}{\mathrm{d}x}$	Partial Derivative	`\per{f}{x_1}`	$\frac{\partial f}{\partial x_1}$
n-th Derivative	`\ner{f}{x}{n}`	$\frac{\mathrm{d}^n f}{\mathrm{d}x^n}$	n-th Partial Derivative	`\pnr{f}{x_1}{n}`	$\frac{\partial^n f}{\partial x_1^n}$
Curl (de)	`\rot`	rot	Curl (en)	`\curl`	curl
Limit (noarg)	`\lims`	lim	Limit	`\lim{n}{\infty}`	$\lim\limits_{n\to\infty}$
Infimum (noarg)	`\infs`	inf	Infimum	`\inf{M}`	$\inf\big(M\big)$
Supremum (noarg)	`\sups`	sup	Supremum	`\sup{M}`	$\sup\big(M\big)$
Limes Inferior (noarg)	`\liminfs`	lim inf	Limes Inferior	`\liminf{n}{\infty}`	$\liminf\limits_{n\to\infty}$
Limes Superior (noarg)	`\limsups`	lim sup	Limes Superior	$\limsup\limits_{n\to\infty}$	$\limsup\limits_{n\to\infty}$
Function Image (de)	`\Bild`	Bild	Function Image (en)	`\Img`	Img
Additional Trigonometric Functions					
Area Sinus hyperbolicus			`\Arsinh`	Arsinh	
Area Cosinus hyperbolicus			`\Arcosh`	Arcosh	
Area Tangens hyperbolicus			`\Artanh`	Artanh	
Area Cotanges hyperbolicus			`\Arcoth`	Arcoth	
Arcus Cotanges	`\arccot`	arccot			
Arcus Secans	`\arcsec`	arcsec	Arcus Cosecans	`\arccsc`	arccsc
Logic					
Bijunction	`\bij`	$\leftrightarrow$			
Equivalent	`\eqv`	$\Leftrightarrow$	Not Equivalent	`\neqv`	$\not\Leftrightarrow$
Right Subjunction	`\subj`	$\rightarrow$	Left Subjunction	`\lsubj`	$\leftarrow$
Not Right Subjunction	`\nsubj`	$\not\rightarrow$	Not Left Subjunction	`\nlsubj`	$\not\leftarrow$
Right Implication	`\implies`	$\Rightarrow$	Left Implication	`\limplies`	$\Leftarrow$
Not Right Implication	`\nimplies`	$\not\Rightarrow$	Not Left Implication	`\nlimplies`	$\not\Leftarrow$
Symbol for True (de)	`\dtrue`	W	Symbol for True (en)	`\etrue`	T
Symbol for False (de)	`\dfalse`	F	Symbol for False (en)	`\efalse`	F
Equations					
Should be equal to	`\feq`	$\stackrel{!}{=}$			
Constants					
Imaginary Unit	`\i`	i	Jimaginary Unit (EE)	`\j`	j
Euler's Number	`\e`	e			
Number Theory					
GCD (de)	`\ggT`	ggT	GCD (en)	`\gcd`	gcd
LCM (de)	`\kgV`	kgV	LCM (en)	`\lcm`	lcm
Signal Transforms					
Laplace Transform	`\ltr{x}`	$\overline{x}$	Z Transform	`\ztr{x}`	$\tilde{x}$
Laplace Transform	`\lap{x}`	$\mathcal{L}\big\{x\big\}(s)$	Laplace Transform (inv)	`\ilap{x}`	$\mathcal{L}^{(-1)}\big\{x\big\}$

Z-Transform	`\zat{x}`	$\mathcal{Z}\{x\}(z)$	Z-Transform (inv)	`\izat{x}`	$\mathcal{Z}^{(-1)}\{x\}$
Fourier Transform	`\frt`	$\overset{\text{FT}}{\longleftrightarrow}$			
Fourier Transform	`\fat{x}`	$\mathcal{F}\{x\}(\omega)$	Fourier Transform (inv)	`\ifat{x}`	$\mathcal{F}^{(-1)}\{x\}$
Fourier Series (de)	`\frr`	$\overset{\text{FR}}{\longleftrightarrow}$	Fourier Series (en)	`\frs`	$\overset{\text{FS}}{\longleftrightarrow}$
DFT	`\dft`	$\overset{\text{DFT}}{\longleftrightarrow}$	DTFT	`\dtft`	$\overset{\text{DTFT}}{\longleftrightarrow}$
Custom TikZ-Symbols for Signal Transforms					
Laplace Transform	`\ltransf`	○—●	Laplace Transform (inv)	`\Ltransf`	●—○
Z Transform	`\ztransf`	□—■	Z Transform (inv)	`\Ztransf`	■—□
Sets					
Natural Numbers	`\N`	$\mathbb{N}$	Integers	`\Z`	$\mathbb{Z}$
Rational Numbers	`\Q`	$\mathbb{Q}$	Irrational Numbers	`\I`	$\mathbb{I}$
Real Numbers	`\R`	$\mathbb{R}$	Complex Numbers	`\C`	$\mathbb{C}$
Set of Primes	`\P`	$\mathbb{P}$	Transcendental Numbers	`\T`	$\mathbb{T}$
General Field (de)	`\K`	$\mathbb{K}$	General Field (en)	`\F`	$\mathbb{F}$

Table 1: All symbols and operators from math.sty

As you might have noticed, some of the entries in the table above feature either (de) or (en). These typically refer to language-dependet Operators. A classic example is the Curl of a Vector-Field. In English, the operator is either $\nabla \times \mathbf{V}$ or curl($\mathbf{V}$). In German however, the cross-prodcut $\nabla \times \mathbf{V}$ ist referred to as *Rotation von* $\mathbf{V}$[1]. Hence the Operator rot($\mathbf{V}$).
There also exist some limits which take no arguments, which is listed with (noarg). This was mostly done to provide a simple text command for just the operator. If you e.g. just want to write: *The limes superior refers to the largest …* and want to use the symbol lim sup in text without any subscript.

0.3 A Word on Tables

Tables in LaTeXcan be quite a pain, especially correct vertical spacing and alignemnt. To avoid maximum frustration, the package `cellspace` is loaded. It allows to define a minimal distance to the top and the bottom of a row. To enable this functionality in your tables, you need to modify your column-list by adding `S` in front of your column type, e.g. `\begin{tabular}{Sc Sl Sr}`. **Note:** If you have `siunitx` loaded[2] you need to write `Cc` instead.
The standard value for space to top/bottom is 4pt. You can change this by modifying the corresponging commands in preamble.sty:

- `\setlength\cellspacetopline` controls the spacing to the top
- `\setlength\cellspacebottomline` controls the spacing to the bottom

preamble also includes the `longtable` package. This allows for tables to perform pagebreak. A pagebreak can be manually inserted by typing `\pagebreak` in the table-contents. In order for this to work, the longtable-environment mustn't be in a table-environment. So wrap your longtable in a center and put the caption as a row element. See `readme.tex` for an example.

[1] Rotation of $\mathbf{V}$
[2] preamble loads this package

0.4 Augmented Matrices and Row Operations

We now support augmented matrices. I took this beautiful solution from Stefan Kottwitz[3]:

```
\makeatletter
\renewcommand*\env@matrix[1][*\c@MaxMatrixCols c]{%
  \hskip -\arraycolsep
  \let\@ifnextchar\new@ifnextchar
  \array{#1}}
\makeatother
```

I found this solution on StackExchange[4]. This modifies the amsmath-matrix environment, such that you can add a column-specification (like for tables) e.g. [cc|c] and after the second column, a line will be drawn. A simple example:

```
\begin{bmatrix}[cc|c]
  m_{11} & m_{12} & b_1\\
  m_{21} & m_{21} & b_2
\end{bmatrix}
```

Produces:

$$\left[\begin{array}{cc|c} m_{11} & m_{12} & b_1 \\ m_{21} & m_{21} & b_2 \end{array}\right]$$

The good part about Kott's solution is, that you can still call `\begin{bmatrix}` and related without any column-specifications, so the following still works:

```
\begin{bmatrix}
  m_{11} & m_{12} & b_1\\
  m_{21} & m_{21} & b_2
\end{bmatrix}
```

Which produces:

$$\begin{bmatrix} m_{11} & m_{12} & b_1 \\ m_{21} & m_{21} & b_2 \end{bmatrix}$$

For Row operations, I found Jake's[5] solution in this[6] thread. It allows you to draw a squiggly arrow with a specified length, which is passed as an argument to the call `\longleadsto{<length>}`.

0.5 Pseudo-Code and Algorithms

`preamble.sty` defines the language `pseudo`, which covers some common keywords for pseudo-code. Here are all defined keywords:

```
        control:
          1) for    2 ) begin 3) end    4) input
          5) if     6 ) then  7) output 8) return
          9) while  10) name
        stack/list operations:
          1) view   2) push   3) pop    4) insert
          5) delete
        basic types:
          1) bool   2) int    3) float  4) double
          5) string 6) data

```

Table 2: Defined Keywords in `pseudo`

To complete the pseudo-code, I found another beautiful solution here[7]. A simple example:

[3] https://tex.stackexchange.com/users/213/stefan-kottwitz[12.3.2021]
[4] https://tex.stackexchange.com/questions/2233/whats-the-best-way-make-an-augmented-coefficient-matrix[12.3.2021]
[5] https://tex.stackexchange.com/users/2552/jake[12.3.2021]
[6] https://tex.stackexchange.com/questions/12678/squiggly-arrows-in-tikz/442036#442036[12.3.2021]
[7] https://tex.stackexchange.com/questions/111116/what-is-the-best-looking-pseudo-code-package

```
name: hanoi
input: index n, Rod a, Rod b, Rod c
hanoi(n, a, b, c):
  if ( n > 0 ) begin
    hanoi(n-1, a, c, b)
    move(a, c)
    hanoi(n-1, b, a, c)
  end

```

Alg 0.5.1: Towers of Hanoi, recursive

The caption-name can be overwritten with a simple `renewcommand{algcapname}{<name>}`. The environment accepts all `lstlisting` arguments as-well as `caption`:

```
\begin{algorithm}[language=pseudo,caption={Towers of Hanoi, recursive}]
  name: hanoi
  input: index n, Rod a, Rod b, Rod c
  hanoi(n, a, b, c):
  if ( n > 0 ) begin
  hanoi(n-1, a, c, b)
  move(a, c)
  hanoi(n-1, b, a, c)
  end
\end{algorithm}
```

0.6 Authors Note

Since I am currently studying Information and Computer Engineering, I've only written macros for corresponding fields (i.e. electrical engineering). So currently there are no neat macros for Chemistry or advanced Physics, etc. Since this repository is public you can Issue a feature request and given some time, it should be implemented in a corresponding style.