

# The `calculator` and `calculus` packages\*

## Scientific calculations with L<sup>A</sup>T<sub>E</sub>X

Robert Fuster  
Universitat Politècnica de València  
`rfuster@mat.upv.es`

2022/09/15

### Abstract

The `calculator` package allows us to use L<sup>A</sup>T<sub>E</sub>X as a calculator, with which we can perform many of the common scientific calculations (with the limitation in accuracy imposed by the T<sub>E</sub>X arithmetic).

This package introduces several new instructions that allow you to do several calculations with integer and decimal numbers using L<sup>A</sup>T<sub>E</sub>X. Apart from add, multiply or divide, we can calculate powers, square roots, logarithms, trigonometric and hyperbolic functions . . .

In addition, the `calculator` package supports some elementary calculations with vectors in two and three dimensions and square  $2 \times 2$  and  $3 \times 3$  matrices.

The `calculus` package adds to the `calculator` package several utilities to use and define various functions and their derivatives, including elementary functions, operations with functions, polar coordinates and vector-valued real functions.

Version 2.0 adds new capabilities to both packages. Specifically, now, `calculator` and `calculus` can evaluate the inverse trigonometric and the inverse hyperbolic functions (so that we can work with all the classic elementary functions), and also can do some additional calculation with vectors (such as the cross product and the angle between two vectors).

Version 2.1 fixes some bugs and calculation problems.<sup>1</sup>

## Contents

### 1 Introduction

The `calculator` package defines some instructions which allow us to realize algebraic operations (and to evaluate elementary functions) in our documents. The operations implemented by the `calculator` package include routines of assignment of variables, arithmetical calculations with real and integer numbers, two and three dimensional vector and matrix arithmetics and the computation of square roots, trigonometrical, exponential, logarithmic and hyperbolic functions. In addition, some important numbers, such as  $\sqrt{2}$ ,  $\pi$  or  $e$ , are predefined.

---

\*This document corresponds to `calculator` v.2.1 and `calculus` v.2.1, dated 2022/09/15.

<sup>1</sup>Thanks to Schmitz Manuel, Thorsten Wolterin, Jim Cline, Schremmer Alain and July Tikhonov.

The name of all these commands is spelled in capital letters (with very few exceptions: the commands `\DEGtoRAD` and `\RADtoDEG` and the control sequences that define special numbers, as `\numberPI`) and, in general, they all need one or more mandatory arguments, the first one(s) of which is(are) number(s) and the last one(s) is(are) the name(s) of the command(s) where the results will be stored.<sup>2</sup> The new commands defined in this way work in any L<sup>A</sup>T<sub>E</sub>X mode.

By example, this instruction

```
\MAX{3}{5}{\solution}
```

stores 5 in the command `\solution`. In a similar way,

```
\FRACTIONSIMPLIFY{10}{12}{\numerator}{\denominator}
```

defines `\numerator` and `\denominator` as 5 i 6, respectively.

The *data* arguments should not be necessarily explicit numbers; it may also consist in commands the value of which is a number. This allows us to chain several calculations, since in the following example:

Ex. 1

$$\begin{aligned} \frac{2.5^2}{\sqrt{12}} + e^{3.4} &= \frac{6.25}{3.4641} + 29.96432 \\ &= 1.80421 + 29.96432 \\ &= 31.76854 \end{aligned}$$

```
% \tempA=2,5^2
\SQUARE{2.5}{\tempA}
% \tempB=sqrt(12)
\SQUAREROOT{12}{\tempB}
% \tempC=exp(3,4)
\EXP{3.4}{\tempC}
% \divisio=\tempA/tempB
\DIVIDE{\tempA}{\tempB}{\divisio}
% \sol=\divisio+\tempC
\ADD{\divisio}{\tempC}{\sol}
\begin{align*}
\frac{2.5^2}{\sqrt{12}}+\mathrm{e}^{3.4}
&= \frac{\tempA}{\tempB}+\tempC \\
&= \divisio+\tempC \\
&=\sol
\end{align*}
```

Observe that, in this example, we have followed exactly the same steps that we would do to calculate  $\frac{2.5^2}{\sqrt{12}} + e^{3.4}$  with a standard calculator: We would calculate the square, the root and the exponential and, finally, we would divide and add the results.

It does not matter if the arguments *results* are or not predefined. But these commands act as declarations, so that its scope is local in environments and groups.

---

<sup>2</sup>Logically, the control sequences that represent special numbers (as `\numberPI`) does not need any argument.

Ex. 2

The `\sol` command contains the square of 5:

$$5^2 = 25$$

Now, the `\sol` command is the square root of 5:

$$\sqrt{5} = 2.23605$$

On having gone out of the `center` environment, the command recovers its previous value: 25

```

\SQUARE{5}\sol
The \texttt{\textbackslash sol}
command contains the square of $5$:
\[5^2=\sol\]
\begin{center}
\SQUAREROOT{5}\sol
Now, the \texttt{\textbackslash sol}
command is the square root of $5$:
\[ \sqrt{5}=\sol\]
\end{center}
On having gone out of the \texttt{center}
environment,
the command recovers its previous value:
\sol

```

The calculus package goes a step further and allows us to define and use in a user-friendly manner various functions and their derivatives.

For exemple, using the calculus package, you can define the  $f(t) = t^2e^t - \cos 2t$  function as follows:

```

% \PRODUCTfunction{\SQUAREfunction}{\EXPfunction}{\tempfunctionA}
% \SCALEVARIABLEfunction{2}{\COSfunction}{\tempfunctionB}
% \SUBTRACTfunction{\tempfunctionA}{\tempfunctionB}{\Ffunction}

```

Then you can compute any value of the new function `\Ffunction` and its derivative: typing

```

\Ffunction{<num>}{\sol}{\Dsol}

```

the values of  $f(num)$  and  $f'(num)$  will be stored in `\sol` and `\Dsol`.

## Part I The calculator package

### 2 Predefined numbers

The calculator package predefines the following numbers:

<code>\numberPI</code>	$3.14159 \approx \pi$	<code>\numberHALFPI</code>	$1.57079 \approx \pi/2$
<code>\numberTHREEHALFPI</code>	$4.71237 \approx 3\pi/2$	<code>\numberTHIRDPI</code>	$1.0472 \approx \pi/3$
<code>\numberQUARTERPI</code>	$0.78539 \approx \pi/4$	<code>\numberFIFTHPI</code>	$0.62831 \approx \pi/5$
<code>\numberSIXTHPI</code>	$0.52359 \approx \pi/6$	<code>\numberTWOPI</code>	$6.28317 \approx 2\pi$
<code>\numberE</code>	$2.71828 \approx e$	<code>\numberINVE</code>	$0.36787 \approx 1/e$
<code>\numberETWO</code>	$7.38902 \approx e^2$	<code>\numberINVETWO</code>	$0.13533 \approx 1/e^2$
<code>\numberLOGTEN</code>	$2.30258 \approx \log 10$		
<code>\numberGOLD</code>	$1.61803 \approx \phi$	<code>\numberINVGOLD</code>	$0.61803 \approx 1/\phi$
<code>\numberSQRTTWO</code>	$1.41421 \approx \sqrt{2}$	<code>\numberSQRTTHREE</code>	$1.73205 \approx \sqrt{3}$
<code>\numberSQRTFIVE</code>	$2.23607 \approx \sqrt{5}$		
<code>\numberCOSXXX</code>	$0.86603 \approx \cos \pi/6$	<code>\numberCOSXLV</code>	$0.70711 \approx \cos \pi/4$

## 3 Operations with numbers

### 3.1 Assignments and comparisons

The first command we describe here is used to store a number in a control sequence. The other two commands in this section determine the maximum and minimum of a pair of numbers.

`\COPY{<num>}{<cmd>}` stores the number *num* to the command *cmd*.

Ex. 3	<code>\COPY{-1.256}{\sol}</code> <code>\sol</code>
-1.256	

`\MAX{<num1>}{<num2>}{<cmd>}` stores in *cmd* the maximum of the numbers *num1* and *num2*.

Ex. 4	<code>\MAX{1.256}{3.214}{\sol}</code> <code>\[\max(1.256,3.214)=\sol\]</code>
$\max(1.256, 3.214) = 3.214$	

`\MIN{<num1>}{<num2>}{<cmd>}` stores in *cmd* the minimum of *num1* and *num2*.

Ex. 5	<code>\MIN{1.256}{3.214}{\sol}</code> <code>\sol</code>
1.256	

### 3.2 Real arithmetic

#### 3.2.1 The four basic operations

The following commands calculate the four arithmetical basic operations.

`\ADD{<num1>}{<num2>}{<cmd>}` Sum of numbers *num1* and *num2*.

Ex. 6	<code>\ADD{1.256}{3.214}{\sol}</code> <code>\$.256+3.214=\sol\$</code>
$1.256 + 3.214 = 4.47$	

`\SUBTRACT{<num1>}{<num2>}{<cmd>}` Difference *num1* - *num2*.

Ex. 7	<code>\SUBTRACT{1.256}{3.214}{\sol}</code> <code>\$.256-3.214=\sol\$</code>
$1.256 - 3.214 = -1.95801$	

`\MULTIPLY{<num1>}{<num2>}{<cmd>}` Product  $num1 \times num2$ .

Ex. 8

$$1.256 \times 3.214 = 4.03677$$

```
\MULTIPLY{1.256}{3.214}{\sol}
$1.256\times3.214=\sol$
```

`\DIVIDE{<num1>}{<num2>}{<cmd>}` Quotient  $num1 / num2$ .<sup>3</sup>

Ex. 9

$$1.256/3.214 = 0.39078$$

```
\DIVIDE{1.256}{3.214}{\sol}
$1.256/3.214=\sol$
```

### 3.2.2 Powers with integer exponent

`\SQUARE{<num>}{<cmd>}` Square of the number  $num$ .

Ex. 10

$$(-1.256)^2 = 1.57751$$

```
\SQUARE{-1.256}{\sol}
$(-1.256)^2=\sol$
```

`\CUBE{<num>}{<cmd>}` Cube of  $num$ .

Ex. 11

$$(-1.256)^3 = -1.98134$$

```
\CUBE{-1.256}{\sol}
$(-1.256)^3=\sol$
```

`\POWER{<num>}{<exp>}{<cmd>}` The  $exp$  power of  $num$ .

The exponent,  $exp$ , must be an integer (if you want to calculate powers with non integer exponents, use the `\EXP` command).

Ex. 12

$$(-1.256)^{-5} = -0.31989$$

$$(-1.256)^5 = -3.1256$$

$$(-1.256)^0 = 1$$

```
\POWER{-1.256}{-5}{\sola}
\POWER{-1.256}{5}{\solb}
\POWER{-1.256}{0}{\solc}
\[
\begin{aligned}
(-1.256)^{-5}&=\sola \\
(-1.256)^5&=\solb \\
(-1.256)^0&=\solc
\end{aligned}
\]
```

---

<sup>3</sup>This command uses a modified version of the division algorithm of Claudio Beccari.

### 3.2.3 Absolute value, integer part and fractional part

`\ABSVALUE{⟨num⟩}{⟨cmd⟩}` Absolute value of  $num$ .

Ex. 13

$$|-1.256| = 1.256$$

```
\ABSVALUE{-1.256}{\sol}
$\left\vert-1.256\right\vert=\sol$
```

`\INTEGERPART{⟨num⟩}{⟨cmd⟩}` Integer part of  $num$ .<sup>4</sup>

Ex. 14

The integer part of 1.256 is 1, but the integer part of  $-1.256$  is  $-2$ .

```
\INTEGERPART{1.256}{\sola}
\INTEGERPART{-1.256}{\solb}
The integer part of $1.256$ is $\sola$,
but the integer part of $-1.256$ is $\solb$.
```

`\FLOOR` is an alias of `\INTEGERPART`.

Ex. 15

The integer part of 1.256 is 1.

```
\FLOOR{1.256}{\sol}
The integer part of $1.256$ is $\sol$.
```

`\FRACTIONALPART{⟨num⟩}{⟨cmd⟩}` Fractional part of  $num$ .<sup>5</sup>

Ex. 16

0.256  
0.744

```
\FRACTIONALPART{1.256}{\sol}
\sol
\FRACTIONALPART{-1.256}{\sol}
\sol
```

### 3.2.4 Truncation and rounding

`\TRUNCATE[⟨n⟩]{⟨num⟩}{⟨cmd⟩}` truncates the number  $num$  to  $n$  decimal places.

`\ROUND[⟨n⟩]{⟨num⟩}{⟨cmd⟩}` rounds the number  $num$  to  $n$  decimal places.<sup>6</sup>

The optional argument  $n$  may be 0, 1, 2, 3 or 4 (the default is 2).<sup>7</sup>

Ex. 17

1  
1.25  
1.2568

```
\TRUNCATE[0]{1.25688}{\sol}
\sol
\TRUNCATE[2]{1.25688}{\sol}
\sol
\TRUNCATE[4]{1.25688}{\sol}
\sol
```

<sup>4</sup>The integer part of  $x$  is the largest integer that is less than or equal to  $x$ .

<sup>5</sup>code modified in version 2.1 (thanks to July Tikhonov who reported a bug and suggested the solution).

<sup>6</sup>code modified in version 2.1 (thanks to Jim Cline and Schremmer Alain who reported a bug).

<sup>7</sup>Note that `\TRUNCATE[0]` is equivalent to `\INTEGERPART` only for non-negative numbers.

Ex. 18

1  
1.26  
1.2569

`\ROUND[0]{1.25688}{\sol}`  
`\sol`

`\ROUND[2]{1.25688}{\sol}`  
`\sol`

`\ROUND[4]{1.25688}{\sol}`  
`\sol`

### 3.3 Integers

The operations described here are subject to the same restrictions as those referring to decimal numbers. In particular, although  $\text{\TeX}$  does not have this restriction in its integer arithmetic, the largest integer that can be used is 16383.

#### 3.3.1 Integer division, quotient and remainder

`\INTEGERDIVISION{num1}{num2}{cmd1}{cmd2}` stores in the `cmd1` and `cmd2` commands the quotient and the remainder of the integer division of the two integers *num1* and *num2*. The remainder is a non-negative number smaller than the divisor.<sup>8</sup>

Ex. 19

$435 = 27 \times 16 + 3$   
 $27 = 435 \times 0 + 27$   
 $-435 = 27 \times (-17) + 24$   
 $435 = -27 \times (-16) + 3$   
 $-435 = -27 \times 17 + 24$

`\INTEGERDIVISION{435}{27}{\sola}{\solb}`  
`$435=27\times\sola+\solb$`

`\INTEGERDIVISION{27}{435}{\sola}{\solb}`  
`$27=435\times\sola+\solb$`

`\INTEGERDIVISION{-435}{27}{\sola}{\solb}`  
`$-435=27\times(\sola)+\solb$`

`\INTEGERDIVISION{435}{-27}{\sola}{\solb}`  
`$435=-27\times(\sola)+\solb$`

`\INTEGERDIVISION{-435}{-27}{\sola}{\solb}`  
`$-435=-27\times\sola+\solb$`

`\INTEGERQUOTIENT{num1}{num2}{cmd}` Integer part of the quotient of *num1* and *num2*. These two numbers are not necessarily integers.

---

<sup>8</sup>The scientific computing systems (such as Matlab, Scilab or Mathematica) do not always return a non-negative residue —especially when the divisor is negative—. However, the most reasonable definition of integer quotient is this one: *the quotient of the division  $D/d$  is the largest number  $q$  for which  $dq \leq D$* . With this definition, the remainder  $r = D - dq$  is a non-negative number.

Ex. 20

16  
0  
-17

```
\INTEGERQUOTIENT{435}{27}{\sol}
\sol
```

```
\INTEGERQUOTIENT{27}{435}{\sol}
\sol
```

```
\INTEGERQUOTIENT{-43.5}{2.7}{\sol}
\sol
```

`\MODULO{<num1>}{<num2>}{<cmd>}` Remainder of the integer division of *num1* and *num2*.

Ex. 21

$435 \equiv 3 \pmod{27}$   
 $-435 \equiv 24 \pmod{27}$

```
\MODULO{435}{27}{\sol}
\[
435 \equiv \sol \pmod{27}
\]
\MODULO{-435}{27}{\sol}
\[
-435 \equiv \sol \pmod{27}
\]
```

### 3.3.2 Greatest common divisor and least common multiple

`\GCD{<num1>}{<num2>}{<cmd>}` Greatest common divisor of the integers *num1* and *num2*.

Ex. 22

$\gcd(435, 27) = 3$

```
\GCD{435}{27}{\sol}
$\gcd(435,27)=\sol$
```

`\LCM{<num1>}{<num2>}{<cmd>}` Least common multiple of *num1* and *num2*.

Ex. 23

$\text{lcm}(435, 27) = 3915$

```
\newcommand{\lcm}{\operatorname{lcm}}
\LCM{435}{27}{\sol}
$\lcm(435,27)=\sol$
```

### 3.3.3 Simplifying fractions

`\FRACTIONSIMPLIFY{<num1>}{<num2>}{<cmd1>}{<cmd2>}` stores in the `<cmd1>` and `<cmd2>` commands the numerator and denominator of the irreducible fraction equivalent to  $\text{num1}/\text{num2}$ .

Ex. 24

$435/27 = 145/9$

```
\FRACTIONSIMPLIFY{435}{27}{\sola}{\solb}
$435/27=\sola/\solb$
```



## 3.4 Elementary functions

### 3.4.1 Square roots

`\SQUAREROOT {<num>}{<cmd>}` Square root of the number *num*.

Ex. 25

$$\sqrt{1.44} = 1.2$$

```
\SQUAREROOT{1.44}{\sol}
$\sqrt{1.44}=\sol$
```

If the argument *num* is negative, the package returns a warning message.

Instead of `\SQUAREROOT`, you can use the alias `\SQRT`.

### 3.4.2 Exponential and logarithm

The `\EXP` and `\LOG` commands compute, by default, exponentials and logarithms of the natural base *e*. They admit, however, an optional argument to choose another base.

`\EXP {<num>}{<cmd>}` Exponential of the number *num*.

Ex. 26

$$\exp(0.5) = 1.64871$$

```
\EXP{0.5}{\sol}
$\exp(0.5)=\sol$
```

The argument *num* must be in the interval  $[-9.704, 9.704]$ .<sup>9</sup>

Moreover, the `\EXP` command accepts an optional argument, to compute expressions such as  $a^x$ :

`\EXP [<num1>]{<num2>}{<cmd>}` Exponential with base *num1* of *num2*. *num1* must be a positive number.

Ex. 27

$$10^{1.3} = 19.95209$$
$$2^{1/3} = 1.25989$$

```
\EXP[10]{1.3}{\sol}
$10^{1.3}=\sol$
```

```
\EXP[2]{0.33333}{\sol}
$2^{1/3}=\sol$
```

`\LOG {<num>}{<cmd>}` logarithm of the number *num*.

Ex. 28

$$\log 0.5 = -0.69315$$

```
\LOG{0.5}{\sol}
$\log 0.5=\sol$
```

---

<sup>9</sup>9.704 is the logarithm of 16383, the largest number that supports the  $\text{T}_{\text{E}}\text{X}$ 's arithmetic.

`\LOG [num1]{num2}{cmd}` Logarithm in base *num1* of *num2*.

Ex. 29

$$\log_{10} 0.5 = -0.30103$$

```
\LOG[10]{0.5}{\sol}
$\log_{10} 0.5=\sol$
```

### 3.4.3 Trigonometric functions

The arguments, in functions `\SIN`, `\COS`, ..., are measured in radians. If you measure angles in degrees (sexagesimal or not), use the `\DEGREESSIN`, `\DEGREESCOS`, ... commands.

`\SIN {num}{cmd}` Sine of *num*.

`\COS {num}{cmd}` Cosine of *num*.

`\TAN {num}{cmd}` Tangent of *num*.

`\COT {num}{cmd}` Cotangent of *num*.

Ex. 30

$$\begin{aligned}\sin \pi/3 &= 0.86601 \\ \cos \pi/3 &= 0.5 \\ \tan \pi/3 &= 1.73201 \\ \cot \pi/3 &= 0.57736\end{aligned}$$

```
\SIN{\numberTHIRDPI}{\sol}
$\sin \pi/3=\sol$
```

```
\COS{\numberTHIRDPI}{\sol}
$\cos \pi/3=\sol$
```

```
\TAN{\numberTHIRDPI}{\sol}
$\tan \pi/3=\sol$
```

```
\COT{\numberTHIRDPI}{\sol}
$\cot \pi/3=\sol$
```

`\DEGREESSIN {num}{cmd}` Sine of *num* sexagesimal degrees.

`\DEGREESCOS {num}{cmd}` Cosine of *num* sexagesimal degrees.

`\DEGREESTAN {num}{cmd}` Tangent of *num* sexagesimal degrees.

`\DEGREESCOT {num}{cmd}` Cotangent of *num* sexagesimal degrees.

Ex. 31

$$\begin{aligned}\sin 60^\circ &= 0.86601 \\ \cos 60^\circ &= 0.49998 \\ \tan 60^\circ &= 1.73201 \\ \cot 60^\circ &= 0.57736\end{aligned}$$

```
\DEGREESSIN{60}{\sol}
$\sin 60^{\text{trm o}}=\sol$
```

```
\DEGREESCOS{60}{\sol}
$\cos 60^{\text{trm o}}=\sol$
```

```
\DEGREESTAN{60}{\sol}
$\tan 60^{\text{trm o}}=\sol$
```

```
\DEGREESCOT{60}{\sol}
$\cot 60^{\text{trm o}}=\sol$
```

The latter commands support an optional argument that allows us to divide the circle in an arbitrary number of *degrees* (not necessarily 360).

`\DEGREESSIN [degrees]{num}{cmd}`

`\DEGREESCOS [degrees]{num}{cmd}`

`\DEGREESTAN [degrees]{num}{cmd}`

`\DEGREESCOT [degrees]{num}{cmd}`

By example, `\DEGREESCOS[400]{50}` computes the cosine of 50 gradians (a right angle has 100 gradians, the whole circle has 400 gradians), which are equivalent to 45 (sexagesimal) degrees or  $\pi/4$  radians. Or to 1 *degree*, if we divide the circle into 8 parts!

Ex. 32

0.70709  
0.70709  
0.7071  
0.70709

```
\DEGREESCOS[400]{50}{\sol}
\sol

\DEGREESCOS{45}{\sol}
\sol

\COS{\numberQUARTERPI}{\sol}
\sol

\DEGREESCOS[8]{1}{\sol}
\sol
```

Moreover, we have a couple of commands to convert between radians and degrees,

`\DEGtoRAD {num}{cmd}` Equivalence in radians of *num* sexagesimal degrees.

`\RADtoDEG {num}{cmd}` Equivalence in sexagesimal degrees of *num* radians.

Ex. 33

1.0472

```
\DEGtoRAD{60}{\sol}
\sol
```

and two other commands to reduce arguments to basic intervals:

`\REDUCERADIANSANGLE {num}{cmd}` Reduces the arc *num* to the interval  $] - \pi, \pi]$ .

`\REDUCEDEGREESANGLE {num}{cmd}` Reduces the angle *num* to the interval  $] - 180, 180]$ .

Ex. 34

3.14159  
90

```
\MULTIPLY{\numberTWOPI}{10}{\TWENTYPI}
\ADD{\numberPI}{\TWENTYPI}{\TWENTYONEPI}
\REDUCERADIANSANGLE{\TWENTYONEPI}{\sol}
\sol

\REDUCEDEGREESANGLE{3690}{\sol}
\sol
```

### 3.4.4 Hyperbolic functions

`\SINH {<num>}{<cmd>}` stores in `<cmd>` the hyperbolic sine of `num`.

`\COSH {<num>}{<cmd>}` Hyperbolic cosine of `num`.

`\TANH {<num>}{<cmd>}` Hyperbolic tangent of `num`.

`\COTH {<num>}{<cmd>}` Hyperbolic cotangent of `num`.

Ex. 35

1.61328  
1.89807  
0.84995  
1.17651

```
\SINH{1.256}{\sol}
\sol
```

```
\COSH{1.256}{\sol}
\sol
```

```
\TANH{1.256}{\sol}
\sol
```

```
\COTH{1.256}{\sol}
\sol
```

### 3.4.5 Inverse trigonometric functions (new in version 2.0)

`\ARCSIN {<num>}{<cmd>}` stores in `<cmd>` the arcsin (inverse of sine) of `num`.

`\ARCCOS {<num>}{<cmd>}` arccos of `num`.

`\ARCTAN {<num>}{<cmd>}` arctan of `num`.

`\ARCCOT {<num>}{<cmd>}` arccot of `num`.

Ex. 36

0.5236  
1.04718  
1.04718  
2.35619

```
\ARCSIN{0.5}{\sol}
\sol
```

```
\ARCCOS{0.5}{\sol}
\sol
```

```
\ARCTAN{\numberSQRTHREE}{\sol}
\sol
```

```
\ARCCOT{-1}{\sol}
\sol
```

### 3.4.6 Inverse hyperbolic functions (new in version 2.0)

`\ARSINH {<num>}{<cmd>}` stores in `<cmd>` the arsinh (inverse of hyperbolic sine) of `num`.

`\ARCOSH {<num>}{<cmd>}` arcosh of `num`.

`\ARTANH {<num>}{<cmd>}` artanh of `num`.

`\ARCOth {<num>}{<cmd>}` arccoth of *num*.

Ex. 37

0.88138  
0  
0.5493  
0.5493

```
\ARSINH{1}{\sol}
\sol
\ARCOSH{1}{\sol}
\sol
\ARTANH{0.5}{\sol}
\sol
\ARCOth{2}{\sol}
\sol
```

## 4 Operations with lengths

`\LENGTHDIVIDE{<length1>}{<length2>}{<cmd>}`

This command divides two lengths and returns a number.

Ex. 38

One inch equals 2.54 centimeters.

```
\LENGTHDIVIDE{1in}{1cm}{\sol}
One inch equals $\sol$ centimeters.
```

Commands `\LENGTHADD` and `\LENGTHSUBTRACT` return the sum and the difference of two lengths (*new in version 2.0*).

`\LENGTHADD{<length1>}{<length2>}{<cmd>}`

`\LENGTHSUBTRACT{<length1>}{<length2>}{<cmd>}`

(`<cmd>` must be a predefined length).

Ex. 39

$1in + 1cm = 100.72273pt.$   
 $1in - 1cm = 43.81725pt.$

```
\newlength{\mylength}
\LENGTHADD{1in}{1cm}{\mylength}
$1in+1cm=\the\mylength$.
\LENGTHSUBTRACT{1in}{1cm}{\mylength}
$1in-1cm=\the\mylength$.
```

## 5 Matrix arithmetic

The `calculator` package defines the commands described below to operate on vectors and matrices. We only work with two or three-dimensional vectors and  $2 \times 2$  and  $3 \times 3$  matrices. Vectors are represented in the form  $(a_1, a_2)$  or  $(a_1, a_2, a_3)$ ; <sup>10</sup> and, in the case of matrices, columns are separated *à la matlab* by semicolons:  $(a_{11}, a_{12}; a_{21}, a_{22})$  or  $(a_{11}, a_{12}, a_{13}; a_{21}, a_{22}, a_{23}; a_{31}, a_{32}, a_{33})$ .

<sup>10</sup>But they are *column* vectors.

## 5.1 Vector operations

### 5.1.1 Assignments

`\VECTORCOPY( $\langle x,y \rangle$ )` (`\cmd1`, `\cmd2`) copy the entries of vector ( $\langle x,y \rangle$ ) to the `\cmd1` and `\cmd2` commands.

`\VECTORCOPY( $\langle x,y,z \rangle$ )` (`\cmd1`, `\cmd2`, `\cmd3`) copy the entries of vector ( $\langle x,y,z \rangle$ ) to the `\cmd1`, `\cmd2` and `\cmd3` commands.

Ex. 40

$$\begin{aligned}(1, -1) \\ (1, -1, 2)\end{aligned}$$

```
\VECTORCOPY(1,-1)(\sola,\solb)
$(\sola,\solb)$
```

```
\VECTORCOPY(1,-1,2)(\sola,\solb,\solc)
$(\sola,\solb,\solc)$
```

### 5.1.2 Vector addition and subtraction

`\VECTORADD( $\langle x_1,y_1 \rangle$ )` (`\langle x_2,y_2 \rangle`) (`\cmd1`, `\cmd2`)

`\VECTORADD( $\langle x_1,y_1,z_1 \rangle$ )` (`\langle x_2,y_2,z_2 \rangle`) (`\cmd1`, `\cmd2`, `\cmd3`)

`\VECTORSUB( $\langle x_1,y_1 \rangle$ )` (`\langle x_2,y_2 \rangle`) (`\cmd1`, `\cmd2`)

`\VECTORSUB( $\langle x_1,y_1,z_1 \rangle$ )` (`\langle x_2,y_2,z_2 \rangle`) (`\cmd1`, `\cmd2`, `\cmd3`)

Ex. 41

$$\begin{aligned}(1, -1, 2) + (3, 5, -1) &= (4, 4, 1) \\ (1, -1, 2) - (3, 5, -1) &= (-2, -6, 3)\end{aligned}$$

```
\VECTORADD(1,-1,2)(3,5,-1)(\sola,\solb,\solc)
$(1,-1,2)+(3,5,-1)=(\sola,\solb,\solc)$
```

```
\VECTORSUB(1,-1,2)(3,5,-1)(\sola,\solb,\solc)
$(1,-1,2)-(3,5,-1)=(\sola,\solb,\solc)$
```

### 5.1.3 Scalar-vector product

`\SCALARVECTORPRODUCT{ $\langle num \rangle$ }` (`\langle x,y \rangle`) (`\cmd1`, `\cmd2`)

`\SCALARVECTORPRODUCT{ $\langle num \rangle$ }` (`\langle x,y,z \rangle`) (`\cmd1`, `\cmd2`, `\cmd3`)

Ex. 42

$$\begin{aligned}2(3, 5) &= (6, 10) \\ 2(3, 5, -1) &= (6, 10, -2)\end{aligned}$$

```
\SCALARVECTORPRODUCT{2}(3,5)(\sola,\solb)
$2(3,5)=(\sola,\solb)$
```

```
\SCALARVECTORPRODUCT{2}(3,5,-1)(%
\sola,\solb,\solc)
$2(3,5,-1)=(\sola,\solb,\solc)$
```

#### 5.1.4 Scalar (dot) product and euclidean norm

`\SCALARPRODUCT( $\langle x1, y1 \rangle$ ) ( $\langle x2, y2 \rangle$ ) { $\langle cmd \rangle$ }`

`\SCALARPRODUCT( $\langle x1, y1, z1 \rangle$ ) ( $\langle x2, y2, z2 \rangle$ ) { $\langle cmd \rangle$ }`

`\DOTPRODUCT` is an alias of `\SCALARPRODUCT` (*new in version 2.0*).

`\VECTORNORM( $\langle x, y \rangle$ ) { $\langle cmd \rangle$ }`

`\VECTORNORM( $\langle x, y, z \rangle$ ) { $\langle cmd \rangle$ }`

Ex. 43

$$\begin{aligned}(1, -1) \cdot (3, 5) &= -2 \\ (1, -1, 2) \cdot (3, 5, -1) &= -4 \\ \|(3, 4)\| &= 5 \\ \|(1, 2, -2)\| &= 3\end{aligned}$$

`\SCALARPRODUCT(1,-1)(3,5){\sol}`

`$(1,-1)\cdot(3,5)=\sol$`

`\DOTPRODUCT(1,-1,2)(3,5,-1){\sol}`

`$(1,-1,2)\cdot(3,5,-1)=\sol$`

`\VECTORNORM(3,4)\sol`

`$$\left|(3,4)\right|=\sol$`

`\VECTORNORM(1,2,-2)\sol`

`$$\left|(1,2,-2)\right|=\sol$`

#### 5.1.5 Vector (cross) product (*new in version 2.0*)

`\VECTORPRODUCT( $\langle x1, y1, z1 \rangle$ ) ( $\langle x2, y2, z2 \rangle$ ) ( $\langle cmd1, cmd2, cmd3 \rangle$ )`

`\CROSSPRODUCT` is an alias of `\VECTORPRODUCT`.

Ex. 44

$$\begin{aligned}(1, -1, 2) \times (3, 5, -1) &= (-9, 7, 8) \\ (1, -1, 2) \times (-3, 3, -6) &= (0, 0, 0)\end{aligned}$$

`\CROSSPRODUCT(1,-1,2)(3,5,-1)%`

`(\sola,\solb,\solc)`

`$(1,-1,2)\times(3,5,-1)=(\sola,\solb,\solc)$`

`\VECTORPRODUCT(1,-1,2)(-3,3,-6)%`

`(\sola,\solb,\solc)`

`$(1,-1,2)\times(-3,3,-6)=(\sola,\solb,\solc)$`

#### 5.1.6 Unit vector parallel to a given vector (normalized vector)

`\UNITVECTOR( $\langle x, y \rangle$ ) ( $\langle cmd1, cmd2 \rangle$ )`

`\UNITVECTOR( $\langle x, y, z \rangle$ ) ( $\langle cmd1, cmd2, cmd3 \rangle$ )`

Ex. 45

$$\begin{aligned}(0.59999, 0.79999) \\ (0.33333, 0.66666, -0.66666)\end{aligned}$$

`\UNITVECTOR(3,4)(\sola,\solb)`

`$(\sola,\solb)$`

`\UNITVECTOR(1,2,-2)(\sola,\solb,\solc)`

`$(\sola,\solb,\solc)$`

### 5.1.7 Absolute value (in each entry of a given vector)

`\VECTORABSVALUE( $\langle x,y \rangle$ ) ( $\langle \text{cmd1}, \text{cmd2} \rangle$ )`

`\VECTORABSVALUE( $\langle x,y,z \rangle$ ) ( $\langle \text{cmd1}, \text{cmd2}, \text{cmd3} \rangle$ )`

Ex. 46

(3, 4)  
(3, 4, 1)

`\VECTORABSVALUE(3,-4) (\sola, \solb)`  
`$(\sola, \solb)$`

`\VECTORABSVALUE(3,-4,-1) (\sola, \solb, \solc)`  
`$(\sola, \solb, \solc)$`

### 5.1.8 Angle between two vectors (new in version 2.0)

`\TWOVECTORSANGLE( $\langle x1,y1 \rangle$ ) ( $\langle x2,y2 \rangle$ ) { $\langle \text{cmd} \rangle$ }`

`\TWOVECTORSANGLE( $\langle x1,y1,z1 \rangle$ ) ( $\langle x2,y2,z2 \rangle$ ) { $\langle \text{cmd} \rangle$ }`

Ex. 47

0.78537 radians (or 44.99837 degrees)  
1.57079 (or 89.99937 degrees)

`\TWOVECTORSANGLE(1,1)(0,1) {\sol}`  
`$(\sol)$ radians`  
`\RADtoDEG{\sol} {\degsol}`  
`(or $(\degsol)$ degrees)`

`\TWOVECTORSANGLE(1,0,0)(0,1,0) {\sol}`  
`$(\sol)$`  
`\RADtoDEG{\sol} {\degsol}`  
`(or $(\degsol)$ degrees)`

## 5.2 Matrix operations

### 5.2.1 Assignments

`\MATRIXCOPY ( $\langle a11,a12;a21,a22 \rangle$ ) ( $\langle \text{cmd11}, \text{cmd12}; \text{cmd21}, \text{cmd22} \rangle$ )`

Use this command to store the matrix  $\begin{bmatrix} a11 & a12 \\ a21 & 22 \end{bmatrix}$  in `\cmm11`, `\cmm12`, `\cmm21`, `\cmm22`.

The analogous  $3 \times 3$  version is

`\MATRIXCOPY ( $\langle a11,a12,a13; [\dots], a33 \rangle$ ) ( $\langle \text{cmd11}, \text{cmd12}, \text{cmd13}; [\dots], \text{cmd33} \rangle$ )`

Ex. 48

$\begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 5 \\ -1 & 1 & 4 \end{bmatrix}$

`\MATRIXCOPY(1, -1, 2;`  
`3, 0, 5;`  
`-1, 1, 4)%`  
`(\sola, \solb, \solc;`  
`\sold, \sole, \self;`  
`\solg, \solh, \soli)`  
`$(\begin{bmatrix}`  
`\sola & \solb & \solc \\`  
`\sold & \sole & \self \\`  
`\solg & \solh & \soli`  
`\end{bmatrix})$`



Henceforth, we will present only the syntax for commands operating with  $2 \times 2$  matrices. In all cases, the syntax is similar if we work with  $3 \times 3$  matrices. In the examples, we will work with either  $2 \times 2$  or  $3 \times 3$  matrices.

### 5.2.2 Transposed matrix

`\TRANPOSEMATRIX (<a11,a12;a21,a22>) (<cmd11,cmd12;cmd21,cmd22>)`

Ex. 49

$$\begin{bmatrix} 1 & -1 \\ 3 & 0 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}$$

```
\TRANPOSEMATRIX(1,-1;3,0)%
(\sola,\solb;\solc,\sold)
 $\begin{bmatrix} 1 & -1 \\ 3 & 0 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ -1 & 0 \end{bmatrix}$ 
\end{bmatrix}
```

### 5.2.3 Matrix addition and subtraction

`\MATRIXADD (<a11,a12;a21,a22>) (<b11,b12;b21,b22>) (<cmd11,cmd12;cmd21,cmd22>)`

`\MATRIXSUB (<a11,a12;a21,a22>) (<b11,b12;b21,b22>) (<cmd11,cmd12;cmd21,cmd22>)`

Ex. 50

$$\begin{bmatrix} 1 & -1 \\ 3 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 5 \\ -3 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 0 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 3 & 5 \\ -3 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -6 \\ 6 & -2 \end{bmatrix}$$

```
\MATRIXADD(1,-1;3,0)(3,5;-3,2)%
(\sola,\solb;\solc,\sold)
 $\begin{bmatrix} 1 & -1 \\ 3 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 5 \\ -3 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 0 & 2 \end{bmatrix}$ 
\end{bmatrix}
```

```
\MATRIXSUB(1,-1;3,0)(3,5;-3,2)%
(\sola,\solb;\solc,\sold)
 $\begin{bmatrix} 1 & -1 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 3 & 5 \\ -3 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -6 \\ 6 & -2 \end{bmatrix}$ 
\end{bmatrix}
```

### 5.2.4 Scalar-matrix product

`\SCALARMATRIXPRODUCT{<num>} (<a11,a12;a21,a22>) (<cmd11,cmd12;cmd21,cmd22>)`

Ex. 51

$$3 \begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 5 \\ -1 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 3 & -3 & 6 \\ 9 & 0 & 15 \\ -3 & 3 & 12 \end{bmatrix}$$

```
\SCALARMATRIXPRODUCT{3}(1,-1,2;
3, 0,5;
-1, 1,4)%
(\sola,\solb,\solc;
\sold,\sole,\solf;
\solg,\solh,\soli)

$3\begin{bmatrix}
1 & -1 & 2 \\
3 & 0 & 5 \\
-1 & 1 & 4
\end{bmatrix}
=\begin{bmatrix}
\sola & \solb & \solc \\
\sold & \sole & \solf \\
\solg & \solh & \soli
\end{bmatrix}$
```

### 5.2.5 Matrix-vector product

`\MATRIXVECTORPRODUCT (<a11,a12;a21,a22>)(<x,y>) (<cmd1,cmd2>)`

Ex. 52

$$\begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} -2 \\ 10 \end{bmatrix}$$

```
\MATRIXVECTORPRODUCT(1,-1;
0, 2)(3,5)(\sola,\solb)

$\begin{bmatrix}
1 & -1 \\
0 & 2
\end{bmatrix}
\begin{bmatrix}
3 \\
5
\end{bmatrix}
=\begin{bmatrix}
\sola \\
\solb
\end{bmatrix}$
```

### 5.2.6 Product of two square matrices

`\MATRIXPRODUCT (<a11,a12;a21,a22>) (<b11,b12;b21,b22>) (<cmd11,cmd12;cmd21,cmd22>)`

Ex. 53

$$\begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 5 \\ -1 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 & 5 & -1 \\ -3 & 2 & -5 \\ 1 & -2 & 3 \end{bmatrix} = \begin{bmatrix} 8 & -1 & 10 \\ 14 & 5 & 12 \\ -2 & -11 & 8 \end{bmatrix}$$

```
\MATRIXPRODUCT(1,-1,2;3,0,5;-1,1,4)%
(3,5,-1;-3,2,-5;1,-2,3)%
(\sola,\solb,\solc;
\sold,\sole,\solf;
\solg,\solh,\soli)
\begin{multline*}
\begin{bmatrix}
1 & -1 & 2 \\ 3 & 0 & 5 \\ -1 & 1 & 4 \end{bmatrix}
\begin{bmatrix}
3 & 5 & -1 \\ -3 & 2 & -5 \\ 1 & -2 & 3 \end{bmatrix} \\
= \begin{bmatrix}
8 & -1 & 10 \\ 14 & 5 & 12 \\ -2 & -11 & 8 \end{bmatrix}
\end{multline*}
```

### 5.2.7 Determinant

`\DETERMINANT (<a11,a12;a21,a22>)` {<cmd>}

Ex. 54

$$\begin{vmatrix} 1 & -1 & 2 \\ 3 & 0 & 5 \\ -1 & 1 & 4 \end{vmatrix} = 18$$

```
\DETERMINANT(1,-1,2;3,0,5;-1,1,4){\sol}
\SpecialUsageIndex{\DETERMINANT}%
$\begin{vmatrix}
1 & -1 & 2 \\ 3 & 0 & 5 \\ -1 & 1 & 4 \end{vmatrix}=\sol$
```

### 5.2.8 Inverse matrix

`\INVERSEMATRIX (<a11,a12;a21,a22>)` (<cmd11,cmd12;cmd21,cmd22>)

Ex. 55

$$\begin{bmatrix} 1 & -1 \\ 3 & 5 \end{bmatrix}^{-1} = \begin{bmatrix} 0.625 & 0.125 \\ -0.375 & 0.125 \end{bmatrix}$$

```
\INVERSEMATRIX(1,-1;3,5){%
\sola,\solb;\solc,\sold)
$\begin{bmatrix}
1 & -1 \\ 3 & 5 \end{bmatrix}^{-1}=
\begin{bmatrix}
0.625 & 0.125 \\ -0.375 & 0.125 \end{bmatrix}
$
```

If the given matrix is singular, the calculator package returns a warning message and the `\cmd11`, ..., commands are marked as undefined.

### 5.2.9 Absolute value (in each entry)

`\MATRIXABSVALUE (<a11,a12;a21,a22>) (<cmd11,cmd12;cmd21,cmd22>)`

Ex. 56

$$\begin{bmatrix} 1 & 1 & 2 \\ 3 & 0 & 5 \\ 1 & 1 & 4 \end{bmatrix}$$

```
\MATRIXABSVALUE(1,-1,2;3,0,5;-1,1,4)%
(\sola,\solb,\solc;
\sold,\sole,\solf;
\solg,\solh,\soli)
$\begin{bmatrix}
\sola & \solb & \solc \\
\sold & \sole & \solf \\
\solg & \solh & \soli
\end{bmatrix}$
```

### 5.2.10 Solving a linear system

`\SOLVELINEARSYSTEM (<a11,a12;a21,a22>)<(b1,b2)> (<cmd1,cmd2>)` solves the linear system  $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$  and stores the solution in `(cmd1,cmd2)`.

Ex. 57

Solving the linear system

$$\begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 5 \\ -1 & 1 & 4 \end{bmatrix} X = \begin{bmatrix} -4 \\ 4 \\ -2 \end{bmatrix}$$

we obtain  $X = \begin{bmatrix} 3 \\ 5 \\ -1 \end{bmatrix}$

```
\SOLVELINEARSYSTEM(1,-1,2;3,0,5;-1,1,4)%
(-4,4,-2)%
(\sola,\solb,\solc)
Solving the linear system
\[
\begin{bmatrix}
1 & -1 & 2 \\
3 & 0 & 5 \\
-1 & 1 & 4
\end{bmatrix} \mathsf{X} = \begin{bmatrix}
-4 \\
4 \\
-2
\end{bmatrix}
\]
we obtain
$\mathsf{X} = \begin{bmatrix}
\sola \\
\solb \\
\solc
\end{bmatrix}$
```

If the given matrix is singular, the package `calculator` returns a warning message. When system is indeterminate, in the bi-dimensional case one of the solutions is computed; if the system is incompatible, then the `\sola`, ..., commands are marked as undefined. For three equations systems, only determinate systems are solved.<sup>11</sup>

<sup>11</sup>This is the only command that does not behave the same way with  $2 \times 2$  and  $3 \times 3$  matrices.

## Part II

# The calculus package

## 6 What is a *function*?

From the point of view of this package, a *function*  $f$  is a pair of formulae: the first one calculates  $f(t)$ ; the other,  $f'(t)$ . Therefore, any function is applied using three arguments: the value of the variable  $t$ , and two command names where  $f(t)$  and  $f'(t)$  will be stored. For example,

```
\SQUAREfunction{<num>}{<\sol>}{<\Dsol>}
```

computes  $f(t) = t^2$  and  $f'(t) = 2t$  (where  $t = num$ ), and stores the results in the commands `\sol` and `\Dsol`.<sup>12</sup>

**Ex. 58**

If  $f(t) = t^2$ , then

$$f(3) = 9 \text{ and } f'(3) = 6$$

```
\SQUAREfunction{3}{\sol}{\Dsol}
If $f(t)=t^2$, then
\[
  f(3)=\sol \mbox{ and } f'(3)=\Dsol
\]
```

For all functions defined here, you must use the following syntax:

```
\functionname{<num>}{<\cmd1>}{<\cmd2>}
```

being  $num$  a number (or a command whose value is a number), and `\cmd1` and `\cmd2` two control sequence names where the values of the function and its derivative (in this number) will be stored.

The key difference between this *functions* and the instructions defined in the *calculator* package is the inclusion of the derivative; for example, the `\SQUARE{3}{\sol}` instruction computes, only, the square power of number 3, while `\SQUAREfunction{3}{\sol}{\Dsol}` finds, also, the corresponding derivative.

## 7 Predefined functions

The *calculus* package predefines the most commonly used elementary functions, and includes several utilities for defining new ones. The predefined functions are the following:

---

<sup>12</sup>Do not spect any control about the existence or differentiability of the function; if the function or the derivative are not well defined, a T<sub>E</sub>X error will occur.

<code>\ZEROfunction</code>	$f(t) = 0$	<code>\ONEfunction</code>	$f(t) = 1$
<code>\IDENTITYfunction</code>	$f(t) = t$	<code>\RECIPROCALfunction</code>	$f(t) = 1/t$
<code>\SQUAREfunction</code>	$f(t) = t^2$	<code>\CUBEfunction</code>	$f(t) = t^3$
<code>\SQRTfunction</code>	$f(t) = \sqrt{t}$		
<code>\EXPfunction</code>	$f(t) = \exp t$	<code>\LOGfunction</code>	$f(t) = \log t$
<code>\COSfunction</code>	$f(t) = \cos t$	<code>\SINfunction</code>	$f(t) = \sin t$
<code>\TANfunction</code>	$f(t) = \tan t$	<code>\COTfunction</code>	$f(t) = \cot t$
<code>\COSHfunction</code>	$f(t) = \cosh t$	<code>\SINHfunction</code>	$f(t) = \sinh t$
<code>\TANHfunction</code>	$f(t) = \tanh t$	<code>\COTHfunction</code>	$f(t) = \coth t$
<code>\HEAVISIDEfunction</code>	$f(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 & \text{si } t \geq 0 \end{cases}$		

The following functions are added in version 2.0 (*new in version 2.0*)

<code>\ARCCOSfunction</code>	$f(t) = \arccos t$	<code>\ARCSINfunction</code>	$f(t) = \arcsin t$
<code>\ARCTANfunction</code>	$f(t) = \arctan t$	<code>\ARCCOTfunction</code>	$f(t) = \operatorname{arccot} t$
<code>\ARCOSHfunction</code>	$f(t) = \operatorname{arcosh} t$	<code>\ARSINHfunction</code>	$f(t) = \operatorname{arsinh} t$
<code>\ARTANHfunction</code>	$f(t) = \operatorname{artanh} t$	<code>\ARCOTHfunction</code>	$f(t) = \operatorname{arcoth} t$

In the following example, we use the `\LOGfunction` function to compute a table of the log function and its derivative.

Ex. 59

$x$	$\log x$	$\log' x$
1	0	1
2	0.69315	0.5
3	1.0986	0.33333
4	1.38629	0.25
5	1.60942	0.2
6	1.79176	0.16666

```

 $\begin{array}{c}
x & \log x & \log' x \\
\LOGfunction{1}{\logx}{\Dlogx} & 1 & \logx & \Dlogx \\
\LOGfunction{2}{\logx}{\Dlogx} & 2 & \logx & \Dlogx \\
\LOGfunction{3}{\logx}{\Dlogx} & 3 & \logx & \Dlogx \\
\LOGfunction{4}{\logx}{\Dlogx} & 4 & \logx & \Dlogx \\
\LOGfunction{5}{\logx}{\Dlogx} & 5 & \logx & \Dlogx \\
\LOGfunction{6}{\logx}{\Dlogx} & 6 & \logx & \Dlogx
\end{array}$ 

```

## 8 Operations with functions

We can define new functions using the following *operations* (the last argument is the name of the new function):

`\CONSTANTfunction{<num>}{<\Function>}` defines `\Function` as the constant function `num`.

Example. Definition of the  $F(t) = 5$  function:

```
\CONSTANTfunction{5}{\F}
```

`\SUMfunction{⟨function1⟩}{⟨function2⟩} {⟨Function⟩}` defines  $\langle Function \rangle$  as the sum of functions  $\langle function1 \rangle$  and  $\langle function2 \rangle$ .

Example. Definition of the  $F(t) = t^2 + t^3$  function:

```
\SUMfunction{\SQUAREfunction}{\CUBEfunction}{\F}
```

`\SUBTRACTfunction{⟨function1⟩}{⟨function2⟩} {⟨Function⟩}` defines  $\langle Function \rangle$  as the difference of functions  $\langle function1 \rangle$  and  $\langle function2 \rangle$ .

Example. Definition of the  $F(t) = t^2 - t^3$  function:

```
\SUBTRACTfunction{\SQUAREfunction}{\CUBEfunction}{\F}
```

`\PRODUCTfunction{⟨function1⟩}{⟨function2⟩} {⟨Function⟩}` defines  $\langle Function \rangle$  as the product of functions  $\langle function1 \rangle$  and  $\langle function2 \rangle$

Example. Definition of the  $F(t) = e^t \cos t$  function:

```
\PRODUCTfunction{\EXPfunction}{\COSfunction}{\F}
```

`\QUOTIENTfunction{⟨function1⟩}{⟨function2⟩} {⟨Function⟩}` defines  $\langle Function \rangle$  as the quotient of functions  $\langle function1 \rangle$  and  $\langle function2 \rangle$ .

Example. Definition of the  $F(t) = e^t / \cos t$  function:

```
\QUOTIENTfunction{\EXPfunction}{\COSfunction}{\F}
```

`\COMPOSITIONfunction{⟨function1⟩}{⟨function2⟩} {⟨Function⟩}` defines  $\langle Function \rangle$  as the composition of functions  $\langle function1 \rangle$  and  $\langle function2 \rangle$ .

Example. Definition of the  $F(t) = e^{\cos t}$  function:

```
\COMPOSITIONfunction{\EXPfunction}{\COSfunction}{\F}
```

(note than `\COMPOSITIONfunction{f}{g}{\F}` means  $\F = f \circ g$ ).

`\SCALEfunction{⟨num⟩}{⟨function⟩} {⟨Function⟩}` defines  $\langle Function \rangle$  as the product of number  $\langle num \rangle$  and function  $\langle function \rangle$ .

Example. Definition of the  $F(t) = 3 \cos t$  function:

```
\SCALEfunction{3}{\COSfunction}{\F}
```

`\SCALEVARIABLEfunction{⟨num⟩}{⟨function⟩} {⟨Function⟩}` scales the variable by factor  $\langle num \rangle$  and then applies the function  $\langle function \rangle$ .

Example. Definition of the  $F(t) = \cos 3t$  function:

```
\SCALEVARIABLEfunction{3}{\COSfunction}{\F}
```

`\POWERfunction{⟨function⟩}{⟨num⟩} {⟨Function⟩}` defines  $\langle Function \rangle$  as the power of function  $\langle function \rangle$  to the exponent  $\langle num \rangle$  (a positive integer). Example. Definition of the  $F(t) = t^5$  function:

```
\POWERfunction{\IDENTITYfunction}{5}{\F}
```

`\LINEARCOMBINATIONfunction{<num1>}{<function1>}{<num2>}{<function2>}{<Function>}` defines `<Function>` as the linear combination of functions `<function1>` and `<function2>` multiplied, respectively, by numbers `num1` and `num2`.

Example. Definition of the  $F(t) = 2t - 3 \cos t$  function:

```
\LINEARCOMBINATIONfunction{2}{\IDENTITYfunction}{-3}{\COSfunction}{\F}
```

By combining properly this operations and the predefined functions, many elementary functions can be defined.

Ex. 60

If	$f(t) = 3t^2 - 2e^{-t} \cos t$	<pre>% exp(-t) \SCALEVARIABLEfunction {-1}{\EXPfunction} {\NEGEXPfunction}</pre>
then	$f(5) = 74.99619$ $f'(5) = 29.99084$	<pre>% exp(-t)cos(t) \PRODUCTfunction {\NEGEXPfunction} {\COSfunction} {\NEGEXPCOSfunction}  % 3t^2-2exp(-t)cos(t) \LINEARCOMBINATIONfunction {3}{\SQUAREfunction} {-2}{\NEGEXPCOSfunction} {\myfunction}  \myfunction{5}{\sol}{\Dsol}</pre>
		<pre>If \[ f(t)=3t^2-2\mathrm{e}^{-t}\cos t \] then \[ \begin{gathered} f(5)=\sol\\ f'(5)=\Dsol \end{gathered} \]</pre>

## 9 Polynomial functions

Although polynomial functions can be defined using linear combinations of power functions, to facilitate our work, the `calculus` package includes the following commands to define more easily the polynomials of 1, 2, and 3 degrees: `\newlpoly` (new *linear* polynomial), `\newqpoly` (new *quadratic* polynomial), and `\newcpoly` (new *cubic* polynomial):

`\newlpoly{<Function>}{<a>}{<b>}` stores the  $p(t) = a + bt$  function in the `<Function>` command.



`\newqpoly{\Function}{a}{b}{c}` stores the  $p(t) = a + bt + ct^2$  function in the `\Function` command.

`\newcpoly{\Function}{a}{b}{c}{d}` stores the  $p(t) = a + bt + ct^2 + dt^3$  function in the `\Function` command.

Ex. 61

$$p'(2) = 8$$

```
% \mypoly=1-x^2+x^3
\newcpoly{\mypoly}{1}{0}{-1}{1}
\mypoly{2}{\sol}{\Dsol}
$p'(2)=\Dsol$
```

These declarations behave similarly to the declaration `\newcommand`: If the name you want to assign to the new function is that of an already defined command, the `calculus` package returns an error message and do not redefines this command. To obtain any alternative behavior, our package includes three other sets of declarations:

`\renewlpoly`, `\renewqpoly`, `\renewcpoly` redefine the already existing command `\Function`. If this command does not exist, then it is not defined and an error message occurs.

`\ensurelpoly`, `\ensureqpoly`, `\ensurecpoly` define a new function. If the command `\Function` already exists, it is not redefined.

`\forcelpoly`, `\forceqpoly`, `\forcecpoly` define a new function. If the command `\Function` already exists, it is redefined.

## 10 Vector-valued functions (or parametrically defined curves)

The instruction

```
\PARAMETRICfunction{\Xfunction}{\Yfunction}{\myvectorfunction}
```

defines the new vector-valued function  $f(t) = (x(t), y(t))$ .

The first and second arguments are a pair of functions already defined and, the third, the name of the new function we define. Once we have defined them, the new vector functions requires five arguments:

```
\myvectorfunction{num}{cmd1}{cmd2}{cmd3}{cmd4}
```

where

- `num` is a number  $t$ ,
- `cmd1` and `cmd2` are two command names where the values of the  $x(t)$  function and its derivative  $x'(t)$  will be stored, and
- `cmd3` and `cmd4` will store  $y(t)$  and  $y'(t)$ .

In short, in this context, a vector function is a pair of scalar functions.

Instead of `\PARAMETRICfunction` we can use the alias `\VECTORfunction`.

Ex. 62

For the  $f(t) = (t^2, t^3)$  function we have

$$f(4) = (16, 64), f'(4) = (8, 48)$$

For the  $f(t)=(t^2,t^3)$  function we have  
`\VECTORfunction`

`{\SQUAREfunction}{\CUBEfunction}{\F}`

`\F{4}{\solx}{\Dsolx}{\soly}{\Dsoly}`

`\[`

`f(4)=(\solx,\soly), f'(4)=(\Dsolx,\Dsoly)`

`\]`

## 11 Vector-valued functions in polar coordinates

The following instruction:

`\POLARfunction{<math>r</math>function}{<math>Polarfunction</math>}`

declares the vector function  $f(\phi) = (r(\phi) \cos \phi, r(\phi) \sin \phi)$ . The first argument is the  $r = r(\phi)$  function, (an already defined function). For example, we can define the *Archimedean spiral*  $r(\phi) = 0,5\phi$ , as follows:

`\SCALEfunction{0.5}{\IDENTITYfunction}{\rfunction}`

`\POLARfunction{\rfunction}{\archimedes}`

## 12 Low-level instructions

Probably, many users of the package will not be interested in the implementation of the commands this package includes. If this is your case, you can ignore this section.

### 12.1 The `\newfunction` declaration and its variants

All the functions predefined by this package use the `\newfunction` declaration. This control sequence works as follows:

`\newfunction{<math>Function</math>}{<math>Instructions to compute \y and \Dy from \t</math>}`

where the second argument is the list of the instructions you need to run to calculate the value of the function  $\y$  and the derivative  $\Dy$  in the  $\t$  point.

For example, if you want to define the  $f(t) = t^2 + e^t \cos t$  function, whose derivative is  $f'(t) = 2t + e^t(\cos t - \sin t)$ , using the high-level instructions we defined earlier, you can write the following instructions:

`\PRODUCTfunction{\EXPfunction}{\COSfunction}{\ffunction}`

`\SUMfunction{\SQUAREfunction}{\ffunction}{\Ffunction}`

But you can also define this function using the `\newfunction` command as follows:

```

\newfunction{\Ffunction}{%
  \SQUARE{\t}{\tempA}           % A=t^2
  \EXP{\t}{\tempB}              % B=e^t
  \COS{\t}{\tempC}              % C=cos(t)
  \SIN{\t}{\tempD}              % D=sin(t)
  \MULTIPLY{2}{\t}{\tempE}      % E=2t
  \MULTIPLY{\tempB}{\tempC}{\tempC} % C=e^t cos(t)
  \MULTIPLY{\tempB}{\tempD}{\tempD} % D=e^t sin(t)
  \ADD{\tempA}{\tempC}{\y}      % y=t^2 + e^t cos(t)
  \ADD{\tempE}{\tempC}{\tempC}  % C=t^2 + e^t cos(t)
  \SUBTRACT{\tempC}{\tempD}{\Dy} % y'=t^2 + e^t cos(t) - e^t sin(t)
}

```

It must be said, however, that the `\newfunction` declaration behaves similarly to `\newcommand` or `\newlpoly`: If the name you want to assign to the new function is that of an already defined command, the calculus package returns an error message and does not redefine this command. To obtain any alternative behavior, our package includes three other versions of the `\newfunction` declarations: the `\renewfunction`, `\ensurefunction` and `\forcefunction` declarations. Each of these declarations behaves differently:

`\newfunction` defines a new function. If the command `\Function` already exists, it is not redefined and an error message occurs.

`\renewfunction` redefines the already existing command `\Function`. If this command does not exist, then it is not defined and an error message occurs.

`\ensurefunction` defines a new function. If the command `\Function` already exists, it is not redefined.

`\forcefunction` defines a new function. If the command `\Function` already exists, it is redefined.

## 12.2 Vector functions and polar coordinates

You can (re)define a vector function  $f(t) = (x(t), y(t))$  using the `\newvectorfunction` declaration or any of its variants `\renewvectorfunction`, `\ensurevectorfunction` and `\forcevectorfunction`:

```

\newvectorfunction{\Function}{\Instructions to compute \x, \Dx, \y and \Dy from \t}

```

For example, you can define the function  $f(t) = (t^2, t^3)$  in the following way:

```

\newvectorfunction{\F}{%
  \SQUARE{\t}{\x}               % x=t^2
  \MULTIPLY{2}{\t}{\Dx}        % x'=2t
  \CUBE{\t}{\y}                % y=t^3
  \MULTIPLY{3}{\x}{\Dy}        % y'=3t^2
}

```

Finally, to define the  $r = r(\phi)$  function, in polar coordinates, we have the declarations `\newpolarfunction`, `\renewpolarfunction`, `\ensurepolarfunction` and `\forcepolarfunction`.

```
\newpolarfunction{\Function}{\Instructions to compute \r and \Dr from \t}
```

For example, you can define the *cardioide* curve  $r(\phi) = 1 + \cos \phi$ , using high level instructions,

```
\SUMfunction{\ONEfunction}{\COSfunction}{\ffunction} % y=1 + cos t
\POLARfunction{\ffunction}{\cardioide}
```

or, with the `\newpolarfunction` declaration,

```
\newpolarfunction{\cardioide}{%
  \COS{\t}{\r}
  \ADD{1}{\r}{\r}          % r=1+cos t
  \SIN{\t}{\Dr}
  \MULTIPLY{-1}{\Dr}{\Dr} % r'=-sin t
}
```

## Part III

# Implementation

### 13 calculator

```
1 (*calculator)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{calculator}[2022/09/15 v.2.1]
```

#### 13.1 Internal lengths and special numbers

`\cctr@lengtha` and `\cctr@lengthb` will be used in internal calculations and comparisons.

```
4 \newdimen\cctr@lengtha
5 \newdimen\cctr@lengthb
```

`\cctr@epsilon` `\cctr@epsilon` will store the closest to zero length in the  $\text{\TeX}$  arithmetic: one scaled point ( $1\text{ sp} = 1/65536\text{ pt}$ ). This means the smallest positive number will be  $0.00002 \approx 1/65536 = 1/2^{16}$ .

```
6 \newdimen\cctr@epsilon
7 \cctr@epsilon=1sp
```

`\cctr@logmaxnum` The largest  $\text{\TeX}$  number is  $16383.99998 \approx 2^{14}$ ; `\cctr@logmaxnum` is the logarithm of this number,  $9.704 \approx \log 16384$ .

```
8 \def\cctr@logmaxnum{9.704}
```

## 13.2 Warning messages

```
9 \def\cctr@Warntruncate#1#2{%
10   \PackageWarning{calculator}%
11     {The optional argument in truncate \MessageBreak
12     must be less than 5 \MessageBreak
13     I copy #1 to #2 \MessageBreak without truncating}}
14
15 \def\cctr@Warnround#1#2{%
16   \PackageWarning{calculator}%
17     {The optional argument in round \MessageBreak
18     must be less than 5 \MessageBreak
19     I copy #1 to #2 \MessageBreak without rounding}}
20
21 \def\cctr@Warndivzero#1#2{%
22   \PackageWarning{calculator}%
23     {Division by 0.\MessageBreak
24     I can't define #1/#2}}
25
26 \def\cctr@Warnnogcd{%
27   \PackageWarning{calculator}%
28     {gcd(0,0) is not well defined}}
29
30 \def\cctr@Warnnopersad#1{%
31   \PackageWarning{calculator}%
32     {The argument in square root\MessageBreak
33     must be non negative\MessageBreak
34     I can't define sqrt(#1)}}
35
36 \def\cctr@Warnnointexp#1#2{%
37   \PackageWarning{calculator}%
38     {The exponent in power function\MessageBreak
39     must be an integer\MessageBreak
40     I can't define #1^#2}}
41
42 \def\cctr@Warnbigarcsin#1{%
43   \PackageWarning{calculator}%
44     {The argument in arcsin\MessageBreak
45     must be a number between -1 and 1\MessageBreak
46     I can't define arcsin(#1)}}
47
48 \def\cctr@Warnbigarccos#1{%
49   \PackageWarning{calculator}%
50     {The argument in arccos\MessageBreak
51     must be a number between -1 and 1\MessageBreak
52     I can't define arccos(#1)}}
53
54 \def\cctr@Warnsmallarcosh#1{%
55   \PackageWarning{calculator}%
56     {The argument in arcosh\MessageBreak
57     must be a number greater or equal than 1\MessageBreak
```

```

58             I can't define arcosh(#1)}}
59
60 \def\cctr@Warnbigartanh#1{%
61     \PackageWarning{calculator}%
62         {The argument in artanh\MessageBreak
63         must be a number between -1 and 1\MessageBreak
64         I can't define artanh(#1)}}
65
66 \def\cctr@Warnsmallarcoth#1{%
67     \PackageWarning{calculator}%
68         {The argument in arcoth\MessageBreak
69         must be a number greater than 1\MessageBreak
70         or smaller than -1\MessageBreak
71         I can't define arcoth(#1)}}
72
73 \def\cctr@Warnsingmatrix#1#2#3#4{%
74     \PackageWarning{calculator}%
75     {Matrix (#1 #2 ; #3 #4) is singular\MessageBreak
76     Its inverse is not defined}}
77
78 \def\cctr@WarnsingTDMatrix#1#2#3#4#5#6#7#8#9{%
79     \PackageWarning{calculator}%
80     {Matrix (#1 #2 #3; #4 #5 #6; #7 #8 #9) is singular\MessageBreak
81     Its inverse is not defined}}
82
83 \def\cctr@WarnIncLinSys{\PackageWarning{calculator}{%
84     Incompatible linear system}}
85
86 \def\cctr@WarnIncTDLinSys{\PackageWarning{calculator}{%
87     Incompatible or indeterminate linear system\MessageBreak
88     For 3x3 systems I can solve only determinate systems}}
89
90 \def\cctr@WarnIndLinSys{\PackageWarning{calculator}{%
91     Indeterminate linear system.\MessageBreak
92     I will choose one of the infinite solutions}}
93
94 \def\cctr@WarnZeroLinSys{\PackageWarning{calculator}{%
95     0x=0 linear system. Every vector is a solution!\MessageBreak
96     I will choose the (0,0) solution}}
97
98 \def\cctr@Warninftan#1{%
99     \PackageWarning{calculator}{%
100         Undefined tangent.\MessageBreak
101         The cosine of #1 is zero and, then,\MessageBreak
102         the tangent of #1 is not defined}}
103
104 \def\cctr@Warninfcotan#1{%
105     \PackageWarning{calculator}{%
106         Undefined cotangent.\MessageBreak
107         The sine of #1 is zero and, then,\MessageBreak

```

```

108             the cotangent of #1 is not defined}}
109
110 \def\cctr@Warninfexp#1{%
111     \PackageWarning{calculator}{%
112         The absolute value of the variable\MessageBreak
113         in the exponential function must be less than
114         \cctr@logmaxnum\MessageBreak
115         (the logarithm of the max number I know)\MessageBreak
116         I can't define exp(#1)}}
117
118 \def\cctr@Warninfexpb#1#2{%
119     \PackageWarning{calculator}{%
120         The base\MessageBreak
121         in the exponential function must be positive.
122         \MessageBreak
123         I can't define #1^(#2)}}
124
125 \def\cctr@Warninflog#1{%
126     \PackageWarning{calculator}{%
127         The value of the variable\MessageBreak
128         in the logarithm function must be positive\MessageBreak
129         I can't define log(#1)}}
130
131 \def\cctr@Warncrossprod(#1)(#2){%
132     \PackageWarning{calculator}{%
133         {Vector product only defined\MessageBreak
134         for 3 dimensional vectors.\MessageBreak
135         I can't define (#1)x(#2)}}
136
137 \def\cctr@Warnnoangle(#1)(#2){%
138     \PackageWarning{calculator}{%
139         {Angle between two vectors only defined\MessageBreak
140         for nonzero vectors.\MessageBreak
141         I can't define an angle between (#1) and (#2)}}}

```

### 13.3 Operations with numbers

#### Assignments and comparisons

`\COPY` `\COPY{<#1>}{<#2>}` defines the `#2` command as the number `#1`.

```
142 \def\COPY#1#2{\edef#2{#1}\ignorespaces}
```

`\GLOBALCOPY` Global version of `\COPY`. The new defined command `#2` is not changed outside groups.

```
143 \def\GLOBALCOPY#1#2{\xdef#2{#1}\ignorespaces}
```

`\@OUTPUTSOL` `\@OUTPUTSOL{<#1>}`: an internal macro to save solutions when a group is closed.

The global c.s. `\cctr@outa` preserves solutions. Whenever we use any temporary parameters in the definition of an instruction, we use a group to ensure the local character of those parameters. The instruction `\@OUTPUTSOL` is a bypass to export the solution.

```
144 \def\@OUTPUTSOL#1{\GLOBALCOPY{#1}{\cctr@outa}\endgroup\COPY{\cctr@outa}{#1}}
```

`\@OUTPUTSOLS` Analogous to `\@OUTPUTSOL`, preserving a pair of solutions.

```

145 \def\@OUTPUTSOLS#1#2{\GLOBALCOPY{#1}{\cctr@outa}
146         \GLOBALCOPY{#2}{\cctr@outb}\endgroup
147         \COPY{\cctr@outa}{#1}\COPY{\cctr@outb}{#2}}

\MAX \MAX{<#1>}{<#2>}{<#3>} defines the #3 command as the maximum of numbers #1 and #2.
148 \def\MAX#1#2#3{%
149     \ifdim #1\p@ < #2\p@
150         \COPY{#2}{#3}\else\COPY{#1}{#3}\fi\ignorespaces}

\MIN \MIN{<#1>}{<#2>}{<#3>} defines the #3 command as the minimum of numbers #1 and #2.
151 \def\MIN#1#2#3{%
152     \ifdim #1\p@ > #2\p@
153         \COPY{#2}{#3}\else\COPY{#1}{#3}\fi\ignorespaces}

```

### Real arithmetic

`\ABSVALUE` `\ABSVALUE{<#1>}{<#2>}` defines the #2 command as the absolute value of number #1.

```

154 \def\ABSVALUE#1#2{%
155     \ifdim #1\p@<z@
156         \MULTIPLY{-1}{#1}{#2}\else\COPY{#1}{#2}\fi}

```

### Product, sum and difference

`\MULTIPLY` `\MULTIPLY{<#1>}{<#2>}{<#3>}` defines the #3 command as the product of numbers #1 and #2.

```

157 \def\MULTIPLY#1#2#3{\cctr@lengtha=#1\p@
158     \cctr@lengtha=#2\cctr@lengtha
159     \edef#3{\expandafter\strip@pt\cctr@lengtha}\ignorespaces}

\ADD \ADD{<#1>}{<#2>}{<#3>} defines the #3 command as the sum of numbers #1 and #2.
160 \def\ADD#1#2#3{\cctr@lengtha=#1\p@
161     \cctr@lengthb=#2\p@
162     \advance\cctr@lengtha by \cctr@lengthb
163     \edef#3{\expandafter\strip@pt\cctr@lengtha}\ignorespaces}

\SUBTRACT \SUBTRACT{<#1>}{<#2>}{<#3>} defines the #3 command as the difference of numbers #1 and #2.
164 \def\SUBTRACT#1#2#3{\ADD{#1}{-#2}{#3}}

```

**Divisions** We define several kinds of *divisions*: the quotient of two real numbers, the integer quotient, and the quotient of two lengths. The basic algorithm is a lightly modified version of the Beccari's division.

`\DIVIDE` `\DIVIDE{<#1>}{<#2>}{<#3>}` defines the #3 command as the quotient of numbers #1 and #2.

```

165 \def\DIVIDE#1#2#3{%
166     \begingroup

```



Absolute values of dividend and divisor

```
167     \ABSVALUE{#1}{\cctr@tempD}
168     \ABSVALUE{#2}{\cctr@tempD}
```

The sign of quotient

```
169     \ifdim#1\p@<\z@\ifdim#2\p@>\z@\COPY{-1}{\cctr@sign}
170         \else\COPY{1}{\cctr@sign}\fi
171     \else\ifdim#2\p@>\z@\COPY{1}{\cctr@sign}
172         \else\COPY{-1}{\cctr@sign}\fi
173     \fi
```

Integer part of quotient

```
174     \@DIVIDE{\cctr@tempD}{\cctr@tempd}{\cctr@tempq}{\cctr@tempr}
175     \COPY{\cctr@tempq.}{\cctr@Q}
```

Fractional part up to five decimal places. `\cctr@ndec` is the number of decimal places already computed.

```
176     \COPY{0}{\cctr@ndec}
177     \@whilenum \cctr@ndec<5 \do{%
```

Each decimal place is calculated by multiplying by 10 the last remainder and dividing it by the divisor. But when the remainder is greater than 1638.3, an overflow occurs, because 16383.99998 is the greatest number. So, instead, we multiply the divisor by 0.1.

```
178         \ifdim\cctr@tempR\p@<1638\p@
179             \MULTIPLY{\cctr@tempR}{10}{\cctr@tempD}
180         \else
181             \COPY{\cctr@tempR}{\cctr@tempD}
182             \MULTIPLY{\cctr@tempD}{0.1}{\cctr@tempd}
183         \fi
184         \@DIVIDE{\cctr@tempD}{\cctr@tempd}{\cctr@tempq}{\cctr@tempR}
185         \COPY{\cctr@Q\cctr@tempq}{\cctr@Q}
186         \ADD{1}{\cctr@ndec}{\cctr@ndec}%
```

Adjust the sign and return the solution.

```
187     \MULTIPLY{\cctr@sign}{\cctr@Q}{#3}
188     \@OUTPUTSOL{#3}
```

`\@DIVIDE` The `\@DIVIDE(<#1>)(<#2>)(<#3>)(<#4>)` command computes  $\#1/\#2$  and returns an integer quotient ( $\#3$ ) and a real remainder ( $\#4$ ).

```
189     \def\@DIVIDE#1#2#3#4{%
190         \@INTEGERDIVIDE{#1}{#2}{#3}
191         \MULTIPLY{#2}{#3}{#4}
192         \SUBTRACT{#1}{#4}{#4}}
```

`\@INTEGERDIVIDE` `\@INTEGERDIVIDE` divides two numbers (not necessarily integer) and returns an integer (this is the integer quotient only for nonnegative integers).

```
193 \def\@INTEGERDIVIDE#1#2#3{%
194     \cctr@lengtha=#1\p@
195     \cctr@lengthb=#2\p@
196     \ifdim\cctr@lengthb=\z@
197         \let#3\undefined
```

```

198         \cctr@Warndivzero#1#2%
199     \else
200         \divide\cctr@lengtha\cctr@lengthb
201         \COPY{\number\cctr@lengtha}{#3}
202     \fi\ignorespaces}

```

**\LENGTHADD** The sum of two lengths. `\LENGTHADD{<#1>}{<#2>}{<#3>}` stores in *#3* the sum of the lengths *#1* and *#2* (*#3* must be a length).

```

203 \def\LENGTHADD#1#2#3{\cctr@lengtha=#1
204     \cctr@lengthb=#2
205     \advance\cctr@lengtha by \cctr@lengthb
206     \setlength{#3}{\cctr@lengtha}\ignorespaces}

```

**\LENGTHSUBTRACT** The difference of two lengths. `\LENGTHSUBTRACT{<#1>}{<#2>}{<#3>}` stores in *#3* the difference of the lengths *#1* and *#2* (*#3* must be a length).

```

207 \def\LENGTHSUBTRACT#1#2#3{%
208     \LENGTHADD{#1}{-#2}{#3}}

```

**\LENGTHDIVIDE** The quotient of two lengths must be a number (not a length). For example, one inch over one centimeter equals 2.54. `\LENGTHDIVIDE{<#1>}{<#2>}{<#3>}` stores in *#3* the quotient of the lengths *#1* and *#2*.

```

209 \def\LENGTHDIVIDE#1#2#3{%
210     \begingroup
211     \cctr@lengtha=#1
212     \cctr@lengthb=#2
213     \edef\cctr@tempa{\expandafter\stripopt\cctr@lengtha}%
214     \edef\cctr@tempb{\expandafter\stripopt\cctr@lengthb}%
215     \DIVIDE{\cctr@tempa}{\cctr@tempb}{#3}
216     \@OUTPUTSOL{#3}}

```

## Powers

**\SQUARE** `\SQUARE{<#1>}{<#2>}` stores *#1* squared in *#2*.

```

217 \def\SQUARE#1#2{\MULTIPLY{#1}{#1}{#2}}

```

**\CUBE** `\CUBE{<#1>}{<#2>}` stores *#1* cubed in *#2*.

```

218 \def\CUBE#1#2{\MULTIPLY{#1}{#1}{#2}\MULTIPLY{#2}{#1}{#2}}

```

**\POWER** `\POWER{<#1>}{<#2>}{<#3>}` stores in *#3* the power  $#1^{#2}$

```

219 \def\POWER#1#2#3{%
220     \begingroup
221     \INTEGERPART{#2}{\cctr@tempexp}
222     \ifdim \cctr@tempexp\p@<#2\p@
223         \cctr@Warnnointexp{#1}{#2}
224         \let#3\undefined
225     \else

```

This ensures that power will be defined only if the exponent is an integer.

```

226     \@POWER{#1}{#2}{#3}\fi\@OUTPUTSOL{#3}}

```

```

227 \def\@POWER#1#2#3{%
228     \begingroup
229     \ifdim #2\p@<\z@

For negative exponents,  $a^n = (1/a)^{-n}$ .

230         \DIVIDE{1}{#1}{\cctr@tempb}
231         \MULTIPLY{-1}{#2}{\cctr@tempc}
232         \@POWER{\cctr@tempb}{\cctr@tempc}{#3}
233     \else
234         \COPY{0}{\cctr@tempa}
235         \COPY{1}{#3}
236         \@whilenum \cctr@tempa<#2 \do {%
237             \MULTIPLY{#1}{#3}{#3}
238             \ADD{1}{\cctr@tempa}{\cctr@tempa}}%
239     \fi\@OUTPUTSOL{#3}}

```

### Integer arithmetic and related things

`\INTEGERDIVISION` `\INTEGERDIVISION{<#1>}{<#2>}{<#3>}{<#4>}` computes the division  $\#1/\#2$  and returns an integer quotient and a positive remainder.

```

240 \def\INTEGERDIVISION#1#2#3#4{%
241     \begingroup
242     \ABSVALUE{#2}{\cctr@tempd}
243     \@DIVIDE{#1}{#2}{#3}{#4}
244     \ifdim #4\p@<\z@
245         \ifdim #1\p@<\z@
246             \ifdim #2\p@<\z@
247                 \ADD{#3}{1}{#3}
248             \else
249                 \SUBTRACT{#3}{1}{#3}
250             \fi
251             \ADD{#4}{\cctr@tempd}{#4}
252     \fi\fi\@OUTPUTSOLS{#3}{#4}}

```

`\MODULO` `\MODULO{<#1>}{<#2>}{<#3>}` returns the remainder of division  $\#1/\#2$ .

```

253 \def\MODULO#1#2#3{%
254     \begingroup
255     \INTEGERDIVISION{#1}{#2}{\cctr@temp}{#3}\@OUTPUTSOL{#3}}

```

`\INTEGERQUOTIENT` `\INTEGERQUOTIENT{<#1>}{<#2>}{<#3>}` returns the integer quotient of division  $\#1/\#2$ .

```

256 \def\INTEGERQUOTIENT#1#2#3{%
257     \begingroup
258     \INTEGERDIVISION{#1}{#2}{#3}{\cctr@temp}\@OUTPUTSOL{#3}}

```

`\INTEGERPART` `\INTEGERPART{<#1>}{<#2>}` returns the integer part of  $\#2$ .

```

259 \def\@INTEGERPART#1.#2.#3)#4{\ifnum #11=1 \COPY{0}{#4}
260     \else \ADD{0}{#1}{#4}\fi}
261 \def\@INTEGERPART#1#2{\expandafter\@INTEGERPART#1..}{#2}}
262 \def\INTEGERPART#1#2{\begingroup
263     \ifdim #1\p@<\z@

```

```

264         \MULTIPLY{-1}{#1}{\cctr@temp}
265         \INTEGERPART{\cctr@temp}{#2}
266         \ifdim #2\p<\cctr@temp\p@
267             \SUBTRACT{-#2}{1}{#2}
268         \else \COPY{-#2}{#2}
269         \fi
270     \else
271         \@INTEGERPART{#1}{#2}
272     \fi\@OUTPUTSOL{#2}}

```

\FLOOR \FLOOR is an alias for \INTEGERPART.

```
273 \let\FLOOR\INTEGERPART
```

\FRACTIONALPART \FRACTIONALPART{<#1>}{<#2>} returns the fractional part of #2.

```

274 \def\@FRACTIONALPART#1.#2.#3)#4{\ifnum #2=1 \COPY{0}{#4}
275     \else \ADD{0}{0.#2}{#4}\fi}
276 \def\@FRACTIONALPART#1#2{\expandafter\@FRACTIONALPART#1..}{#2}}
277 \def\FRACTIONALPART#1#2{\begingroup
278     \ifdim #1\p<\z@
279         \INTEGERPART{#1}{\cctr@tempA}
280         \SUBTRACT{#1}{\cctr@tempA}{#2}
281     \else
282         \@FRACTIONALPART{#1}{#2}
283     \fi\@OUTPUTSOL{#2}}
284

```

\TRUNCATE \TRUNCATE[<#1>]{<#2>}{<#3>} truncates #2 to #1 (0, 1, 2 (default), 3 or 4) digits.

```

285 \def\TRUNCATE{\@ifnextchar [\@TRUNCATE\@TRUNCATE}
286 \def\@TRUNCATE#1#2{\@TRUNCATE[2]{#1}{#2}}
287 \def\@TRUNCATE[#1]#2#3{%
288     \begingroup
289     \ifdim #1\p@ > 4\p@ \cctr@Warntruncate{#2}{\noexpand#3} \COPY{#2}{#3}
290     \else
291         \INTEGERPART{#2}{\cctr@tempa}
292         \ifdim \cctr@tempa\p@ = #2\p@
293             \expandafter\@TRUNCATE\cctr@tempa.00000.)[#1]{#3}
294         \else
295             \expandafter\@TRUNCATE#200000.)[#1]{#3}
296         \fi\fi
297     \@OUTPUTSOL{#3}}
298
299 \def\@TRUNCATE#1.#2#3#4#5#6.#7)[#8]#9{%
300     \ifcase #8
301         \COPY{#1}{#9}
302     \or\COPY{#1.#2}{#9}
303     \or\COPY{#1.#2#3}{#9}
304     \or\COPY{#1.#2#3#4}{#9}
305     \or\COPY{#1.#2#3#4#5}{#9}
306     \fi}

```

`\ROUND` `\ROUND[⟨#1⟩]{⟨#2⟩}{⟨#3⟩}` rounds #2 to #1 (0, 1, 2 (default), 3 or 4) digits.

```

307 \def\ROUND{\@ifnextchar[{\@ROUND\@ROUND}
308 \def\@ROUND#1#2{\@ROUND[2]{#1}{#2}}
309 \def\@ROUND[#1]#2#3{%
310   \begingroup
311   \ifdim #1\p@ > 4\p@ \cctr@Warnround{#2}{\noexpand#3} \COPY{#2}{#3}
312   \else
313   \INTEGERPART{#2}{\cctr@tempa}
314   \ifdim \cctr@tempa\p@ = #2\p@
315     \expandafter\@@ROUND\cctr@tempa.00000.)[#1]{#3}
316   \else
317     \expandafter\@@ROUND#200000.)[#1]{#3}
318   \fi
319   \fi
320   \@OUTPUTSOL{#3}}
321
322 \def\@@ROUND#1.#2#3#4#5#6.#7)[#8]#9{%
323   \ifcase #8
324     \COPY{#1}{#9} \ifnum #2>4 \ADD{#1}{1}{\cctr@tempq}\COPY{\cctr@tempq}{#9} \fi
325     \or\COPY{#1.#2}{#9} \ifnum #3>4 \ADD{#2}{1}\cctr@tempq\COPY{#1}{\cctr@tempq}
326       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}\cctr@tempq\COPY{0}{\cctr@tempq}
327       \COPY{\cctr@tempq.\cctr@tempq}{#9}
328     \fi
329     \or\COPY{#1.#2#3}{#9} \ifnum #4>4 \ADD{#3}1\cctr@tempq\COPY{#2}{\cctr@tempq}\COPY{#1}{\cctr@tempq}
330       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}{\cctr@tempq}\COPY{0}{\cctr@tempq}
331       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}\cctr@tempq\COPY{0}{\cctr@tempq}
332       \COPY{\cctr@tempq.\cctr@tempq\cctr@tempq}{#9}
333     \fi
334     \or\COPY{#1.#2#3#4}{#9} \ifnum #5>4 \ADD{#4}1\cctr@tempq\COPY{#3}{\cctr@tempq}\COPY{#2}{\cctr@tempq}
335       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}{\cctr@tempq}\COPY{0}{\cctr@tempq}
336       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}\cctr@tempq\COPY{0}{\cctr@tempq}
337       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}\cctr@tempq\COPY{0}{\cctr@tempq}
338       \COPY{\cctr@tempq.\cctr@tempq\cctr@tempq\cctr@tempq}{#9}
339     \fi
340     \or\COPY{#1.#2#3#4#5}{#9} \ifnum #6>4 \ADD{#5}1\cctr@tempq\COPY{#4}{\cctr@tempq}\COPY{#3}{\cctr@tempq}
341       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}{\cctr@tempq}\COPY{0}{\cctr@tempq}
342       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}\cctr@tempq\COPY{0}{\cctr@tempq}
343       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}\cctr@tempq\COPY{0}{\cctr@tempq}
344       \ifnum\cctr@tempq=10\ADD{\cctr@tempq}{1}\cctr@tempq\COPY{0}{\cctr@tempq}
345       \COPY{\cctr@tempq.\cctr@tempq\cctr@tempq\cctr@tempq\cctr@tempq}{#9}
346   \fi}

```

`\GCD` `\GCD{⟨#1⟩}{⟨#2⟩}{⟨#3⟩}` Greatest common divisor, using the Euclidean algorithm

```

347 \def\GCD#1#2#3{%
348   \begingroup
349   \ABSVALUE{#1}{\cctr@tempa}
350   \ABSVALUE{#2}{\cctr@tempb}
351   \MAX{\cctr@tempa}{\cctr@tempb}{\cctr@tempc}
352   \MIN{\cctr@tempa}{\cctr@tempb}{\cctr@tempa}
353   \COPY{\cctr@tempc}{\cctr@tempb}

```

```

354     \ifnum \cctr@tempa = 0
355         \ifnum \cctr@tempb = 0
356             \cctr@Warnnogcd
357             \let#3\undefined
358         \else
359             \COPY{\cctr@tempb}{#3}
360         \fi
361     \else

```

Euclidean algorithm: if  $c \equiv b \pmod{a}$  then  $\gcd(b, a) = \gcd(a, c)$ . Iterating this property, we obtain  $\gcd(b, a)$  as the last nonzero residual.

```

362     \@whilenum \cctr@tempa > \z@ \do {%
363         \COPY{\cctr@tempa}{#3}%
364         \MODULO{\cctr@tempb}{\cctr@tempa}{\cctr@tempc}%
365         \COPY{\cctr@tempa}{\cctr@tempb}%
366         \COPY{\cctr@tempc}{\cctr@tempa}%
367     \fi\ignorespaces\@OUTPUTSOL{#3}}

```

`\LCM` `\LCM{<#1>}{<#2>}{<#3>}` Least common multiple.

```

368 \def\LCM#1#2#3{%
369     \GCD{#1}{#2}{#3}%
370     \ifx #3\undefined \COPY{0}{#3}
371     \else
372         \DIVIDE{#1}{#3}{#3}
373         \MULTIPLY{#2}{#3}{#3}
374         \ABSVALUE{#3}{#3}
375     \fi}

```

`\FRACTIONSIMPLIFY` `\FRACTIONSIMPLIFY{<#1>}{<#2>}{<#3>}{<#4>}` Fraction simplification:  $\#3/\#4$  is the irreducible fraction equivalent to  $\#1/\#2$ .

```

376 \def\FRACTIONSIMPLIFY#1#2#3#4{%
377     \ifnum #1=\z@
378         \COPY{0}{#3}\COPY{1}{#4}
379     \else
380         \GCD{#1}{#2}{#3}%
381         \DIVIDE{#2}{#3}{#4}
382         \DIVIDE{#1}{#3}{#3}
383         \ifnum #4<0 \MULTIPLY{-1}{#4}{#4}\MULTIPLY{-1}{#3}{#3}\fi
384     \fi\ignorespaces}

```

## Elementary functions

### Square roots

`\SQUAREROOT` `\SQUAREROOT{<#1>}{<#2>}` defines  $\#2$  as the square root of  $\#1$ , using the Newton's method:  

$$x_{n+1} = x_n - (x_n^2 - \#1)/(2x_n).$$

```

385 \def\SQUAREROOT#1#2{%
386     \begingroup
387     \ifdim #1\p@ = \z@
388         \COPY{0}{#2}

```

```

389     \else
390     \ifdim #1\p@ < \z@
391     \let#2\undefined
392     \cctr@Warnnopersad{#1}%
393     \else
We take #1 as the initial approximation.
394     \COPY{#1}{#2}
\cctr@lengthb will be the difference of two successive iterations.
We start with \cctr@lengthb=5\p@ to ensure almost one iteration.
395     \cctr@lengthb=5\p@
Successive iterations
396     \@whilenum \cctr@lengthb>\cctr@epsilon \do {%
Copy the actual approximation to \cctr@tempw
397     \COPY{#2}{\cctr@tempw}
398     \DIVIDE{#1}{\cctr@tempw}{\cctr@tempz}
399     \ADD{\cctr@tempw}{\cctr@tempz}{\cctr@tempz}
400     \DIVIDE{\cctr@tempz}{2}{\cctr@tempz}
Now, \cctr@tempz is the new approximation.
401     \COPY{\cctr@tempz}{#2}
Finally, we store in \cctr@lengthb the difference of the two last approximations, finishing the
loop.
402     \SUBTRACT{#2}{\cctr@tempw}{\cctr@tempw}
403     \cctr@lengthb=\cctr@tempw\p@%
404     \ifnum
405     \cctr@lengthb<\z@ \cctr@lengthb=-\cctr@lengthb
406     \fi}
407     \fi\fi\@OUTPUTSOL{#2}}

```

`\SQRT` `\SQRT` is an alias for `\SQUAREROOT`.

```
408 \let\SQRT\SQUAREROOT
```

**Trigonometric functions** For a variable close enough to zero, the sine and tangent functions are computed using some continued fractions. Then, all trigonometric functions are derived from well-known formulas.

`\SIN` `\SIN{<#1>}{<#2>}`. Sine of #1.

```

409 \def\SIN#1#2{%
410     \begingroup
Exact sine for  $t \in \{\pi/2, -\pi/2, 3\pi/2\}$ 
411     \ifdim #1\p@=-\numberHALFPI\p@ \COPY{-1}{#2}
412     \else
413     \ifdim #1\p@=\numberHALFPI\p@ \COPY{1}{#2}
414     \else
415     \ifdim #1\p@=\numberTHREEHALFPI\p@ \COPY{-1}{#2}
416     \else

```

If  $|t| > \pi/2$ , change  $t$  to a smaller value.

```

417         \ifdim#1\p@<-\numberHALFPI\p@
418             \ADD{#1}{\numberTWOPI}{\cctr@tempb}
419             \SIN{\cctr@tempb}{#2}
420         \else
421             \ifdim #1\p@<\numberHALFPI\p@
Compute the sine.
422             \@BASICSINE{#1}{#2}
423         \else
424             \ifdim #1\p@<\numberTHREEHALFPI\p@
425                 \SUBTRACT{\numberPI}{#1}{\cctr@tempb}
426                 \SIN{\cctr@tempb}{#2}
427             \else
428                 \SUBTRACT{#1}{\numberTWOPI}{\cctr@tempb}
429                 \SIN{\cctr@tempb}{#2}
430 \fi\fi\fi\fi\fi\fi\@OUTPUTSOL{#2}

```

`\@BASICSINE \@BASICSINE{<#1>}{<#2>}` applies this approximation:

$$\sin x = \frac{x}{1 + \frac{x^2}{2 \cdot 3 - x^2 + \frac{2 \cdot 3x^2}{4 \cdot 5 - x^2 + \frac{4 \cdot 5x^2}{6 \cdot 7 - x^2 + \dots}}}}$$

```

431 \def\@BASICSINE#1#2{%
432     \begingroup
433     \ABSVALUE{#1}{\cctr@tempa}
Exact sine of zero
434     \ifdim\cctr@tempa\p@=\z@ \COPY{0}{#2}
435     \else
For  $t$  very close to zero,  $\sin t \approx t$ .
436     \ifdim \cctr@tempa\p@<0.009\p@\COPY{#1}{#2}
437     \else

```

Compute the continued fraction.

```

438         \SQUARE{#1}{\cctr@tempa}
439         \DIVIDE{\cctr@tempa}{42}{#2}
440         \SUBTRACT{1}{#2}{#2}
441         \MULTIPLY{#2}{\cctr@tempa}{#2}
442         \DIVIDE{#2}{20}{#2}
443         \SUBTRACT{1}{#2}{#2}
444         \MULTIPLY{#2}{\cctr@tempa}{#2}
445         \DIVIDE{#2}{6}{#2}
446         \SUBTRACT{1}{#2}{#2}
447         \MULTIPLY{#2}{#1}{#2}
448 \fi\fi\@OUTPUTSOL{#2}

```



`\COS` `\COS{⟨#1⟩}{⟨#2⟩}`. Cosine of  $\#1$ :  $\cos t = \sin(t + \pi/2)$ .

```

449 \def\COS#1#2{%
450     \begingroup
451     \ADD{\numberHALFPI}{#1}{\ctr@tempc}
452     \SIN{\ctr@tempc}{#2}\@OUTPUTSOL{#2}}

```

`\TAN` `\TAN{⟨#1⟩}{⟨#2⟩}`. Tangent of  $\#1$ .

```

453 \def\TAN#1#2{%
454     \begingroup
455     Tangent is infinite for  $t = \pm\pi/2$ 
456     \ifdim #1\p@=-\numberHALFPI\p@
457         \ctr@WarninfTan{#1}
458         \let#2\undefined
459     \else
460         \ifdim #1\p@=\numberHALFPI\p@
461             \ctr@WarninfTan{#1}
462             \let#2\undefined
463         \else
464             If  $|t| > \pi/2$ , change  $t$  to a smaller value.
465             \ifdim #1\p@<-\numberHALFPI\p@
466                 \ADD{#1}{\numberPI}{\ctr@tempb}
467                 \TAN{\ctr@tempb}{#2}
468             \else
469                 \ifdim #1\p@<\numberHALFPI\p@
470                     Compute the tangent.
471                     \@BASICTAN{#1}{#2}
472                 \else
473                     \SUBTRACT{#1}{\numberPI}{\ctr@tempb}
474                     \TAN{\ctr@tempb}{#2}
475                 \fi\fi\fi\@OUTPUTSOL{#2}}

```

`\@BASICTAN` `\@BASICTAN{⟨#1⟩}{⟨#2⟩}` applies this approximation:

$$\tan x = \frac{1}{\frac{1}{x} - \frac{3}{x} \frac{1}{\frac{5}{x} - \frac{7}{x} \frac{1}{\frac{9}{x} - \frac{11}{x} \dots}}}}$$

```

473 \def\@BASICTAN#1#2{%
474     \begingroup
475     \ABSVALUE{#1}{\ctr@tempa}
476     Exact tangent of zero.
477     \ifdim\ctr@tempa\p@=\z@ \COPY{0}{#2}
478     \else

```

For  $t$  very close to zero,  $\tan t \approx t$ .

```

478         \ifdim\cctr@tempa\p@<0.04\p@
479             \COPY{#1}{#2}
480         \else
    Compute the continued fraction.
481             \DIVIDE{#1}{11}{#2}
482             \DIVIDE{9}{#1}{\cctr@tempa}
483             \SUBTRACT{\cctr@tempa}{#2}{#2}
484             \DIVIDE{1}{#2}{#2}
485             \DIVIDE{7}{#1}{\cctr@tempa}
486             \SUBTRACT{\cctr@tempa}{#2}{#2}
487             \DIVIDE{1}{#2}{#2}
488             \DIVIDE{5}{#1}{\cctr@tempa}
489             \SUBTRACT{\cctr@tempa}{#2}{#2}
490             \DIVIDE{1}{#2}{#2}
491             \DIVIDE{3}{#1}{\cctr@tempa}
492             \SUBTRACT{\cctr@tempa}{#2}{#2}
493             \DIVIDE{1}{#2}{#2}
494             \DIVIDE{1}{#1}{\cctr@tempa}
495             \SUBTRACT{\cctr@tempa}{#2}{#2}
496             \DIVIDE{1}{#2}{#2}
497         \fi\fi\@OUTPUTSOL{#2}}

```

`\COT` `\COT{<#1>}{<#2>}`. Cotangent of  $\#1$ : If  $\cos t = 0$  then  $\cot t = 0$ ; if  $\tan t = 0$  then  $\cot t = \infty$ . Otherwise,  $\cot t = 1/\tan t$ .

```

498 \def\COT#1#2{%
499     \begingroup
500     \COS{#1}{#2}
501     \ifdim #2\p@ = \z@
502     \COPY{0}{#2}
503     \else
504     \TAN{#1}{#2}
505     \ifdim #2\p@ = \z@
506     \cctr@Warninfcotan{#1}
507     \let#2\undefined
508     \else
509     \DIVIDE{1}{#2}{#2}
510     \fi\fi\@OUTPUTSOL{#2}}

```

`\DEGtoRAD` `\DEGtoRAD{<#1>}{<#2>}`. Convert degrees to radians.

```

511 \def\DEGtoRAD#1#2{\DIVIDE{#1}{57.29578}{#2}}

```

`\RADtoDEG` `\RADtoDEG{<#1>}{<#2>}`. Convert radians to degrees.

```

512 \def\RADtoDEG#1#2{\MULTIPLY{#1}{57.29578}{#2}}

```

`\REDUCERADIANSANGLE` Reduces to the trigonometrically equivalent arc in  $]-\pi, \pi]$ .

```

513 \def\REDUCERADIANSANGLE#1#2{%
514     \COPY{#1}{#2}
515     \ifdim #1\p@ < -\numberPI\p@

```

```

516             \ADD{#1}{\numberTWOPI}{#2}
517             \REDUCERADIANSANGLE{#2}{#2}
518         \fi
519     \ifdim #1\p@ > \numberPI\p@
520         \SUBTRACT{#1}{\numberTWOPI}{#2}
521         \REDUCERADIANSANGLE{#2}{#2}
522     \fi
523     \ifdim #1\p@ = -180\p@ \COPY{\numberPI}{#2} \fi}

```

`\REDUCEDEGREESANGLE` Reduces to the trigonometrically equivalent angle in  $[-180, 180]$ .

```

524 \def\REDUCEDEGREESANGLE#1#2{%
525     \COPY{#1}{#2}
526     \ifdim #1\p@ < -180\p@
527         \ADD{#1}{360}{#2}
528         \REDUCEDEGREESANGLE{#2}{#2}
529     \fi
530     \ifdim #1\p@ > 180\p@
531         \SUBTRACT{#1}{360}{#2}
532         \REDUCEDEGREESANGLE{#2}{#2}
533     \fi
534     \ifdim #1\p@ = -180\p@ \COPY{180}{#2} \fi}

```

**Trigonometric functions in degrees** Four next commands compute trigonometric functions in *degrees*. By default, a circle has 360 degrees, but we can use an arbitrary number of divisions using the optional argument of these commands.

```

\DEGREESSIN \DEGREESSIN[⟨#1⟩]{⟨#2⟩}{⟨#3⟩}. Sine of #2 degrees.
535 \def\DEGREESSIN{\@ifnextchar[\@@DEGREESSIN\@DEGREESSIN}

\DEGREESCOS \DEGREESCOS[⟨#1⟩]{⟨#2⟩}{⟨#3⟩}. Cosine of #2 degrees.
536 \def\DEGREESCOS{\@ifnextchar[\@@DEGREESCOS\@DEGREESCOS}

\DEGREESTAN \DEGREESTAN[⟨#1⟩]{⟨#2⟩}{⟨#3⟩}. Tangent of #2 degrees.
537 \def\DEGREESTAN{\@ifnextchar[\@@DEGREESTAN\@DEGREESTAN}

\DEGREESCOT \DEGREESCOT[⟨#1⟩]{⟨#2⟩}{⟨#3⟩}. Cotangent of #2 degrees.
538 \def\DEGREESCOT{\@ifnextchar[\@@DEGREESCOT\@DEGREESCOT}

```

`\@DEGREESSIN` `\@DEGREESSIN` computes the sine in sexagesimal *degrees*.

```

539 \def\@DEGREESSIN#1#2{%
540     \begingroup
541     \ifdim #1\p@=-90\p@ \COPY{-1}{#2}
542     \else
543         \ifdim #1\p@=90\p@ \COPY{1}{#2}
544         \else
545             \ifdim #1\p@=270\p@ \COPY{-1}{#2}
546             \else
547                 \ifdim#1\p@<-90\p@
548                     \ADD{#1}{360}{\cctr@tempb}

```

```

549         \DEGREESSIN{\cctr@tempb}{#2}
550     \else
551         \ifdim #1\p@<90\p@
552             \DEGtoRAD{#1}{\cctr@tempb}
553             \@BASICSINE{\cctr@tempb}{#2}
554         \else
555             \ifdim #1\p@<270\p@
556                 \SUBTRACT{180}{#1}{\cctr@tempb}
557                 \DEGREESSIN{\cctr@tempb}{#2}
558             \else
559                 \SUBTRACT{#1}{360}{\cctr@tempb}
560                 \DEGREESSIN{\cctr@tempb}{#2}
561     \fi\fi\fi\fi\fi\fi\@OUTPUTSOL{#2}}

```

\@DEGREESCOS \@DEGREESCOS computes the cosine in sexagesimal *degrees*.

```

562 \def\@DEGREESCOS#1#2{%
563     \begingroup
564     \ADD{90}{#1}{\cctr@tempc}
565     \DEGREESSIN{\cctr@tempc}{#2}\@OUTPUTSOL{#2}}

```

\@DEGREESTAN \@DEGREESTAN computes the tangent in sexagesimal *degrees*.

```

566 \def\@DEGREESTAN#1#2{%
567     \begingroup
568     \ifdim #1\p@=-90\p@
569         \cctr@Warninftan{#1}
570         \let#2\undefined
571     \else
572         \ifdim #1\p@=90\p@
573             \cctr@Warninftan{#1}
574             \let#2\undefined
575         \else
576             \ifdim #1\p@<-90\p@
577                 \ADD{#1}{180}{\cctr@tempb} \DEGREESTAN{\cctr@tempb}{#2}
578             \else
579                 \ifdim #1\p@<90\p@
580                     \DEGtoRAD{#1}{\cctr@tempb}
581                     \@BASICTAN{\cctr@tempb}{#2}
582                 \else
583                     \SUBTRACT{#1}{180}{\cctr@tempb}
584                     \DEGREESTAN{\cctr@tempb}{#2}
585             \fi\fi\fi\fi\@OUTPUTSOL{#2}}

```

\@DEGREESCOT \@DEGREESCOT computes the cotangent in sexagesimal *degrees*.

```

586 \def\@DEGREESCOT#1#2{%
587     \begingroup
588     \DEGREESCOS{#1}{#2}
589     \ifdim #2\p@ = \z@
590         \COPY{0}{#2}
591     \else
592         \DEGREESTAN{#1}{#2}

```

```

593     \ifdim #2\p@ = \z@
594     \ctr@Warninfcotan{#1}
595     \let#2\undefined
596     \else
597     \DIVIDE{1}{#2}{#2}
598     \fi\fi\@OUTPUTSOL{#2}}

```

For an arbitrary number of *degrees*, we normalise to 360 degrees and, then, call the former functions.

`\@@DEGREESSIN` `\@@DEGREESSIN` computes the sine. A circle has *#1 degrees*.

```

599 \def\@@DEGREESSIN[#1]#2#3{\@CONVERTDEG{#1}{#2}
600     \@DEGREESSIN{\@DEGREES}{#3}}

```

`\@@DEGREESCOS` `\@@DEGREESCOS` computes the sine. A circle has *#1 degrees*.

```

601 \def\@@DEGREESCOS[#1]#2#3{\@CONVERTDEG{#1}{#2}
602     \DEGREESCOS{\@DEGREES}{#3}}

```

`\@@DEGREESTAN` `\@@DEGREESTAN` computes the sine. A circle has *#1 degrees*.

```

603 \def\@@DEGREESTAN[#1]#2#3{\@CONVERTDEG{#1}{#2}
604     \DEGREESTAN{\@DEGREES}{#3}}

```

`\@@DEGREESCOT` `\@@DEGREESCOT` computes the sine. A circle has *#1 degrees*.

```

605 \def\@@DEGREESCOT[#1]#2#3{\@CONVERTDEG{#1}{#2}
606     \DEGREESCOT{\@DEGREES}{#3}}

```

`\@CONVERTDEG` `\@CONVERTDEG` normalises to sexagesimal degrees.

```

607 \def\@CONVERTDEG#1#2{\DIVIDE{#2}{#1}{\@DEGREES}
608     \MULTIPLY{\@DEGREES}{360}{\@DEGREES}}

```

### Exponential functions

`\EXP` `\EXP` [*#1*] [*#2*] [*#3*] computes the exponential  $#3 = #1^{#2}$ . Default for *#1* is number e.

```

609 \def\EXP{\@ifnextchar[\@@EXP\@EXP}

```

`\@@EXP` `\@@EXP` [*#1*] [*#2*] [*#3*] computes  $#3 = #1^{#2}$

```

610 \def\@@EXP[#1]#2#3{
611     \begingroup
        #1 must be a positive number.
612     \ifdim #1\p@<\ctr@epsilon
613         \ctr@Warninfexpb{#1}{#2}
614         \let#3\undefined
615     \else
         $a^b = \exp(b \log a)$ .
616         \LOG{#1}{\ctr@log}
617         \MULTIPLY{#2}{\ctr@log}{\ctr@log}
618         \EXP{\ctr@log}{#3}
619     \fi\@OUTPUTSOL{#3}}

```

`\@EXP \@EXP{<#1>}{<#2>}` computes  $\#3 = e^{\#2}$

```

620 \def\@EXP#1#2{%
621     \begingroup
622     \ABSVALUE{#1}{\cctr@absval}
    If |t| is greater than \cctr@logmaxnum then exp t is too large.
623     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
624         \cctr@Warninfexp{#1}
625         \let#2\undefined
626     \else
627         \ifdim #1\p@ < \z@

```

We call `\@BASICEXP` when  $t \in [-6, 3]$ . Otherwise we use the equality  $\exp t = (\exp t/2)^2$ .

```

628         \ifdim #1\p@ > -6.00002\p@
629             \@BASICEXP{#1}{#2}
630         \else
631             \DIVIDE{#1}{2}{\cctr@expt}
632             \@EXP{\cctr@expt}{\cctr@expy}
633             \SQUARE{\cctr@expy}{#2}
634         \fi
635     \else
636         \ifdim #1\p@ < 3.00002\p@
637             \@BASICEXP{#1}{#2}
638         \else
639             \DIVIDE{#1}{2}{\cctr@expt}
640             \@EXP{\cctr@expt}{\cctr@expy}
641             \SQUARE{\cctr@expy}{#2}
642         \fi
643 \fi\fi\@OUTPUTSOL{#2}}

```

`\@BASICEXP \@BASICEXP{<#1>}{<#2>}` applies this approximation:

$$\exp x \approx 1 + \frac{2x}{2 - x + \frac{x^2/6}{1 + \frac{x^2/60}{1 + \frac{x^2/140}{1 + \frac{x^2/256}{1 + \frac{x^2}{396}}}}}}$$

```

644 \def\@BASICEXP#1#2{%
645     \begingroup
646     \SQUARE{#1}\cctr@tempa
647     \DIVIDE{\cctr@tempa}{396}{#2}
648     \ADD{1}{#2}{#2}
649     \DIVIDE\cctr@tempa{#2}{#2}
650     \DIVIDE{#2}{256}{#2}
651     \ADD{1}{#2}{#2}
652     \DIVIDE\cctr@tempa{#2}{#2}
653     \DIVIDE{#2}{140}{#2}

```

```

654     \ADD{1}{#2}{#2}
655     \DIVIDE\cctr@tempa{#2}{#2}
656     \DIVIDE{#2}{60}{#2}
657     \ADD{1}{#2}{#2}
658     \DIVIDE\cctr@tempa{#2}{#2}
659     \DIVIDE{#2}{6}{#2}
660     \ADD{2}{#2}{#2}
661     \SUBTRACT{#2}{#1}{#2}
662     \DIVIDE{#1}{#2}{#2}
663     \MULTIPLY{2}{#2}{#2}
664     \ADD{1}{#2}{#2}\@OUTPUTSOL{#2}}

```

### Hyperbolic functions

`\COSH` `\COSH`. Hyperbolic cosine:  $\cosh t = (\exp t + \exp(-t))/2$ .

```

665 \def\COSH#1#2{%
666     \begingroup
667     \ABSVALUE{#1}{\cctr@absval}
668     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
669         \cctr@Warninfexp{#1}
670         \let#2\undefined
671     \else
672         \EXP{#1}{\cctr@exp}
673         \MULTIPLY{-1}{#1}{\cctr@minust}
674         \EXP{\cctr@minust}{\cctr@expminusx}
675         \ADD{\cctr@exp}{\cctr@expminusx}{#2}
676         \DIVIDE{#2}{2}{#2}
677     \fi \@OUTPUTSOL{#2}}

```

`\SINH` `\SINH`. Hyperbolic sine:  $\sinh t = (\exp t - \exp(-t))/2$ .

```

678 \def\SINH#1#2{%
679     \begingroup
680     \ABSVALUE{#1}{\cctr@absval}
681     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
682         \cctr@Warninfexp{#1}
683         \let#2\undefined
684     \else
685         \EXP{#1}{\cctr@exp}
686         \MULTIPLY{-1}{#1}{\cctr@minust}
687         \EXP{\cctr@minust}{\cctr@expminusx}
688         \SUBTRACT{\cctr@exp}{\cctr@expminusx}{#2}
689         \DIVIDE{#2}{2}{#2}
690     \fi \@OUTPUTSOL{#2}}

```

`\TANH` `\TANH`. Hyperbolic tangent:  $\tanh t = \sinh t / \cosh t$ .

```

691 \def\TANH#1#2{%
692     \begingroup
693     \ABSVALUE{#1}{\cctr@absval}
694     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
695         \cctr@Warninfexp{#1}

```

```

696         \let#2\undefined
697     \else
698         \SINH{#1}{\cctr@tanhnnum}
699         \COSH{#1}{\cctr@tanhdn}
700         \DIVIDE{\cctr@tanhnnum}{\cctr@tanhdn}{#2}
701     \fi\@OUTPUTSOL{#2}

```

**\COTH** **\COTH**. Hyperbolic cotangent  $\coth t = \cosh t / \sinh t$ .

```

702 \def\COTH#1#2{%
703     \begingroup
704     \ABSVALUE{#1}{\cctr@absval}
705     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
706         \cctr@Warninffexp{#1}
707         \let#2\undefined
708     \else
709         \SINH{#1}{\cctr@tanhdn}
710         \COSH{#1}{\cctr@tanhnnum}
711         \DIVIDE{\cctr@tanhnnum}{\cctr@tanhdn}{#2}
712     \fi\@OUTPUTSOL{#2}

```

### Logarithm

**\LOG** **\LOG** $\langle\#1\rangle\langle\#2\rangle\langle\#3\rangle$  computes the logarithm  $\#3 = \log_{\#1} \#2$ . Default for  $\#1$  is number  $e$ .

```

713 \def\LOG{\@ifnextchar[\@LOG\@LOG}

```

**\@LOG** **\@LOG** $\langle\#1\rangle\langle\#2\rangle$  computes  $\#2 = \log \#1$

```

714 \def\@LOG#1#2{%
715     \begingroup
716     The argument  $t$  must be positive.
717     \ifdim #1\p@<\cctr@epsilon
718         \cctr@Warninfflog{#1}
719         \let#2\undefined
720     \else
721         \ifdim #1\p@ > \numberETWO\p@
722             If  $t > e^2$ ,  $\log t = \log e + \log(t/e) = 1 + \log(t/e)$ 
723             \DIVIDE{#1}{\numberE}{\cctr@ae}
724             \@LOG{\cctr@ae}{#2}
725             \ADD{1}{#2}{#2}
726         \else
727             \ifdim #1\p@ < 1\p@
728                 If  $t < 1$ ,  $\log t = \log(1/e) + \log(te) = -1 + \log(te)$ 
729                 \MULTIPLY{\numberE}{#1}{\cctr@ae}
730                 \LOG{\cctr@ae}{#2}
731                 \SUBTRACT{#2}{1}{#2}
732             \else

```



For  $t \in [1, e^2]$  we call `\@@BASICLOG`.

```
730 \@@BASICLOG{#1}{#2}
731 \fi\fi\fi\@OUTPUTSOL{#2}}
```

`\@LOG` `\@LOG[<#1>]{<#2>}{<#3>}` computes  $\#3 = \log_{\#1} \#2 = \log(\#2)/\log(\#1)$

```
732 \def\@LOG[#1]#2#3{\begingroup
733 \@LOG{#1}{\ctr@loga}
734 \@LOG{#2}{\ctr@logx}
735 \DIVIDE{\ctr@logx}{\ctr@loga}{#3}\@OUTPUTSOL{#3}}
```

`\@BASICLOG` `\@BASICLOG{<#1>}{<#2>}` applies the Newton's method to calculate  $x = \log t$ :

$$x_{n+1} = x_n + \frac{t}{e^{x_n}} - 1$$

```
736 \def\@BASICLOG#1#2{\begingroup
737 % We take $\textit{#1}$-1$ as the initial approximation.
738 % \begin{macrocode}
739 \SUBTRACT{#1}{1}{\ctr@tempw}
    We start with \ctr@lengthb=5\p@ to ensure almost one iteration.
740 \ctr@lengthb=5\p@%
741 \ctr@epsilon=2\ctr@epsilon%
```

Successive iterations

```
742 \@whilenum \ctr@lengthb>\ctr@epsilon \do {%
743 \COPY{\ctr@tempw}{\ctr@tempoldw}
744 \EXP{\ctr@tempw}{\ctr@tempxw}
745 \DIVIDE{#1}{\ctr@tempxw}{\ctr@tempxw}
746 \ADD{\ctr@tempw}{\ctr@tempxw}{\ctr@tempw}
747 \SUBTRACT{\ctr@tempw}{1}{\ctr@tempw}
748 \SUBTRACT{\ctr@tempw}{\ctr@tempoldw}{\ctr@tempdif}
749 \ctr@lengthb=\ctr@tempdif\p@%
750 \ifnum
    \ctr@lengthb<\z@ \ctr@lengthb=-\ctr@lengthb
752 \fi}%
753 \COPY{\ctr@tempw}{#2}\@OUTPUTSOL{#2}}
```

### Inverse trigonometric functions

`\ARCSIN` `\ARCSIN{<#1>}{<#2>}` defines  $\#2$  as the arcsin of  $\#1$ , using the Newton's method:  $x_{n+1} = x_n - (\sin x_n - \#1)/(\cos x_n)$ .

```
754 \def\ARCSIN#1#2{%
755 \begingroup
756 \ifdim #1\p@ = \z@
757 \COPY{0}{#2}
758 \else
759 \ifdim #1\p@ = 1\p@
760 \COPY{\numberHALFPI}{#2}
761 \else
762 \ifdim #1\p@ = -1\p@
```

```

763         \COPY{-\numberHALFPI}{#2}
764     \else
765         \ifdim #1\p@ > 1\p@
766             \let#2\undefined
767             \ctr@Warnbigarcsin{#1}
768         \else
769             \ifdim #1\p@ < -1\p@
770                 \let#2\undefined
771                 \ctr@Warnbigarcsin{#1}
772             \else
If  $x$  is close to 1 we use  $\arcsin x = \pi/2 - 2 \arcsin \sqrt{(1-x)/2}$ 
773         \ifdim #1\p@ > 0.89\p@
774             \SUBTRACT{1}{#1}{\ctr@tempx}
775             \DIVIDE{\ctr@tempx}{2}{\ctr@tempx}
776             \SQRT{\ctr@tempx}{\ctr@tempxx}
777             \ARCSIN{\ctr@tempxx}{#2}
778             \MULTIPLY{2}{#2}{#2}
779             \SUBTRACT{\numberHALFPI}{#2}{#2}
780         \else
Symmetrically, for  $x$  close to  $-1$ ,  $\arcsin x = -\pi/2 + 2 \arcsin \sqrt{(1+x)/2}$ 
781         \ifdim #1\p@ < -0.89\p@
782             \ADD{1}{#1}{\ctr@tempx}
783             \DIVIDE{\ctr@tempx}{2}{\ctr@tempx}
784             \SQRT{\ctr@tempx}{\ctr@tempxx}
785             \ARCSIN{\ctr@tempxx}{#2}
786             \MULTIPLY{2}{#2}{#2}
787             \SUBTRACT{#2}{\numberHALFPI}{#2}
788         \else
We take  $\#1$  as the initial approximation.
789         \COPY{#1}{#2}
If  $-0.4 \leq t \leq 0.4$  then  $\arcsin x \approx x$  is a good approximation. Else, we apply the Newton method
790         \ABSVALUE{#1}{\ctr@tempy}
791         \ifdim \ctr@tempy\p@ < 0.04\p@
792         \else
\ctr@lengthb will be the difference of two successive iterations, and \ctr@tempoldy,
\ctr@tempy will be the two last iterations.
We start with \ctr@lengthb=5\p@ and \ctr@tempy=16383 to ensure almost one iteration.
793         \ctr@lengthb=5\p@
794         \COPY{16383}{\ctr@tempy}
Successive iterations
795         \@whilenum \ctr@lengthb>\ctr@epsilon \do {%
Copy the actual approximation to \ctr@tempw
796         \COPY{#2}{\ctr@tempw}
797         \COPY{\ctr@tempy}{\ctr@tempoldy}
798         \SIN{\ctr@tempw}{\ctr@tempz}
799         \SUBTRACT{\ctr@tempz}{#1}{\ctr@tempz}

```

```

800             \COS{\cctr@tempw}{\cctr@tempy}
801             \DIVIDE{\cctr@tempz}{\cctr@tempy}{\cctr@tempz}
802             \SUBTRACT{\cctr@tempw}{\cctr@tempz}{\cctr@tempz}

```

Now, `\cctr@tempz` is the new approximation.

```

803             \COPY{\cctr@tempz}{#2}

```

Finally, we store in `\cctr@lengthb` the difference of the two last approximations, finishing the loop.

```

804             \SUBTRACT{#2}{\cctr@tempw}{\cctr@tempy}
805             \ABSVALUE{\cctr@tempy}{\cctr@tempy}
806             \cctr@lengthb=\cctr@tempy\p@%
807             \ifdim\cctr@tempy\p@=\cctr@tempoldy\p@
808                 \cctr@lengthb=\z@
809             \fi}\fi\fi\fi\fi\fi\fi\fi\fi\@OUTPUTSOL{#2}}

```

`\ARCCOS` `\ARCCOS{<#1>}{<#2>}` defines `#2` as the arccos of `#1`, using the well know relation  $\arccos x = \pi/2 - \arcsin x$ .

```

810 \def\ARCCOS#1#2{%
811     \begingroup
812     \ifdim #1\p@ = \z@
813         \COPY{\numberHALFPI}{#2}
814     \else
815         \ifdim #1\p@ = 1\p@
816             \COPY{0}{#2}
817         \else
818             \ifdim #1\p@ = -1\p@
819                 \COPY{\numberPI}{#2}
820             \else
821                 \ifdim #1\p@ > 1\p@
822                     \let#2\undefined
823                     \cctr@Warnbigarccos{#1}
824                 \else
825                     \ifdim #1\p@ < -1\p@
826                         \let#2\undefined
827                         \cctr@Warnbigarccos{#1}
828                     \else
829                         \ARCSIN{#1}{#2}
830                         \SUBTRACT{\numberHALFPI}{#2}{#2}
831                     \fi}\fi\fi\fi\fi\@OUTPUTSOL{#2}}

```

`\ARCTAN` `\ARCTAN{<#1>}{<#2>}`. arctan of `#1`.

```

832 \def\ARCTAN#1#2{%
833     \begingroup
834     If  $|t| > 1$ , compute arctan  $x$  using  $\arctan x = \text{sign}(x)\pi/2 - \arctan(1/x)$ .
835     \ifdim#1\p@<-1\p@
836         \DIVIDE{1}{#1}{\cctr@tempb}
837         \ARCTAN{\cctr@tempb}{#2}
838         \SUBTRACT{-\numberHALFPI}{#2}{#2}
839     \else

```

```

839             \ifdim#1\p@>1\p@
840             \DIVIDE{1}{#1}{\ctr@tempb}
841             \ARCTAN{\ctr@tempb}{#2}
842             \SUBTRACT{\numberHALFPI}{#2}{#2}
843         \else
For  $-1 \leq x \leq 1$  call \@BASICARCTAN.
844             \@BASICARCTAN{#1}{#2}
845         \fi
846     \fi\@OUTPUTSOL{#2}}

```

\@BASICARCTAN \@BASICARCTAN{<#1>}{<#2>} applies this approximation:

$$\arctan x = \frac{x}{1 + \frac{x^2}{3 + \frac{(2x)^2}{5 + \frac{(3x)^2}{7 + \frac{(4x)^2}{9 + \dots}}}}}$$

```

847 \def\@BASICARCTAN#1#2{%
848     \begingroup
Exact arctan of zero
849     \ifdim#1\p@=\z@ \COPY{0}{#2}
850     \else
Compute the continued fraction.
851         \SQUARE{#1}{\ctr@tempa}
852         \MULTIPLY{64}{\ctr@tempa}{#2}
853         \ADD{15}{#2}{#2}
854         \DIVIDE{\ctr@tempa}{#2}{#2}
855         \MULTIPLY{49}{#2}{#2}
856         \ADD{13}{#2}{#2}
857         \DIVIDE{\ctr@tempa}{#2}{#2}
858         \MULTIPLY{36}{#2}{#2}
859         \ADD{11}{#2}{#2}
860         \DIVIDE{\ctr@tempa}{#2}{#2}
861         \MULTIPLY{25}{#2}{#2}
862         \ADD{9}{#2}{#2}
863         \DIVIDE{\ctr@tempa}{#2}{#2}
864         \MULTIPLY{16}{#2}{#2}
865         \ADD{7}{#2}{#2}
866         \DIVIDE{\ctr@tempa}{#2}{#2}
867         \MULTIPLY{9}{#2}{#2}
868         \ADD{5}{#2}{#2}
869         \DIVIDE{\ctr@tempa}{#2}{#2}
870         \MULTIPLY{4}{#2}{#2}
871         \ADD{3}{#2}{#2}
872         \DIVIDE{\ctr@tempa}{#2}{#2}
873         \ADD{1}{#2}{#2}

```

```

874          \DIVIDE{#1}{#2}{#2}
875          \fi \@OUTPUTSOL{#2}}

```

`\ARCCOT` `\ARCCOT{<#1>}{<#2>}` defines `#2` as the arccot of `#1`, using the well know relation  $\operatorname{arccot} x = \pi/2 - \arctan x$ .

```

876 \def\ARCCOT#1#2{%
877     \begingroup
878     \ARCTAN{#1}{#2}
879     \SUBTRACT{\numberHALFPI}{#2}{#2}
880     \@OUTPUTSOL{#2}}

```

### Inverse hyperbolic functions

`\ARSINH` `\ARSINH{<#1>}{<#2>}`. Inverse hyperbolic sine of `#1`:  $\operatorname{arsinh} x = \log(x + \sqrt{1 + x^2})$

```

881 \def\ARSINH#1#2{%
882     \begingroup
883     \SQUARE{#1}{\cctr@tempa}
884     \ADD{1}{\cctr@tempa}{\cctr@tempa}
885     \SQRT{\cctr@tempa}{\cctr@tempb}
886     \ADD{#1}{\cctr@tempb}{\cctr@tempb}
887     \LOG\cctr@tempb{#2}
888     \@OUTPUTSOL{#2}}

```

`\ARCOSH` `\ARCOSH{<#1>}{<#2>}`. Inverse hyperbolic sine of `#1`:  $\operatorname{arcosh} x = \log(x + \sqrt{x^2 - 1})$

```

889 \def\ARCOSH#1#2{%
890     \begingroup
891     If  $x < 1$ , this function is no defined
892     \ifdim#1\p<1\p@
893         \let#2\undefined
894         \cctr@Warnsmallarcosh{#1}
895     \else
896         \SQUARE{#1}{\cctr@tempa}
897         \SUBTRACT{\cctr@tempa}{1}{\cctr@tempa}
898         \SQRT{\cctr@tempa}{\cctr@tempb}
899         \ADD{#1}{\cctr@tempb}{\cctr@tempb}
900         \LOG\cctr@tempb{#2}
901     \fi \@OUTPUTSOL{#2}}

```

`\ARTANH` `\ARTANH{<#1>}{<#2>}`. Inverse hyperbolic tangent of `#1`:  $\operatorname{artanh} x = \frac{1}{2} \log((1 + x) - \log(1 - x))$

```

901 \def\ARTANH#1#2{%
902     \begingroup
903     If  $|x| \geq 1$ , this function is no defined
904     \ifdim#1\p<-0.99998\p@
905         \let#2\undefined
906         \cctr@Warnbigartanh{#1}
907     \else
908         \ifdim#1\p>0.99998\p@
909             \let#2\undefined
910         \fi
911     \fi \@OUTPUTSOL{#2}}

```

```

909         \cctr@Warnbigartanh{#1}
910     \else
911         \COPY{#1}{\cctr@tempa}
912         \ADD1\cctr@tempa\cctr@tempb
913         \SUBTRACT1\cctr@tempa\cctr@tempc
914         \LOG\cctr@tempb\cctr@tempB
915         \LOG\cctr@tempc\cctr@tempC
916         \SUBTRACT\cctr@tempB\cctr@tempC{#2}
917         \DIVIDE{#2}{2}{#2}
918     \fi
919 \fi\@OUTPUTSOL{#2}}

```

`\ARCOTH` `\ARCOTH{<#1>}{<#2>}`. Inverse hyperbolic cotangent of `#1`:  

$$\operatorname{arcoth} x = \operatorname{sign}(x) \frac{1}{2} \log((x+1) - \log(x-1))$$

```

920 \def\ARCOTH#1#2{%
921     \begingroup
922     If  $|x| \leq 1$ , this function is no defined
923     \ifdim#1\p@>-0.99998\p@
924         \ifdim#1\p@<0.99998\p@
925             \let#2\undefined
926             \cctr@Warnsmallarcoth{#1}
927         \else
928             \ifdim#1\p@>\p@
929                 For  $x > 1$ , calculate  $\operatorname{arcoth} x = \frac{1}{2} \log((x+1) - \log(x-1))$ 
930                 \COPY{#1}{\cctr@tempa}
931                 \ADD1\cctr@tempa\cctr@tempb
932                 \SUBTRACT\cctr@tempa1\cctr@tempc
933                 \LOG\cctr@tempb\cctr@tempB
934                 \LOG\cctr@tempc\cctr@tempC
935                 \SUBTRACT\cctr@tempB\cctr@tempC{#2}
936                 \DIVIDE{#2}{2}{#2}
937             \else
938                 \fi
939             \fi
940             For  $x < -1$ , calculate  $-\operatorname{artanh}(-x)$ 
941             \MULTIPLY{-1}{#1}{\cctr@tempa}
942             \ARCOTH{\cctr@tempa}{#2}
943             \COPY{-#2}{#2}
944         \fi\@OUTPUTSOL{#2}}

```

## 13.4 Matrix arithmetics

### Vector operations

`\VECTORSIZE` The *size* of a vector is 2 or 3. `\VECTORSIZE{<#1>}{<#2>}` stores in `#2` the size of `(<#1>)`.

Almost all vector commands needs to know the vector size.

```

943 \def\VECTORSIZE(#1)#2{\expandafter\@VECTORSIZE(#1,,){#2}}

```

```

944 \def\@VECTORSIZE(#1,#2,#3,#4)#5{\ifx#3$\COPY{2}{#5}
945 \else\COPY{3}{#5}\fi\ignorespaces}

\VECTORCOPY \VECTORCOPY(\langle#1,#2\rangle)(\langle#3,#4\rangle) stores #1 and #2 in #3 and #4.
\VECTORCOPY(\langle#1,#2,#3\rangle)(\langle#4,#5#6\rangle) stores #1, #2 and #3 in #4 and #5 and #6.
946 \def\@@VECTORCOPY(#1,#2)(#3,#4){%
947 \COPY{#1}{#3}\COPY{#2}{#4}}
948
949 \def\@@@VECTORCOPY(#1,#2,#3)(#4,#5,#6){%
950 \COPY{#1}{#4}\COPY{#2}{#5}\COPY{#3}{#6}}
951
952 \def\VECTORCOPY(#1)(#2){%
953 \VECTORSIZE(#1){\ctr@size}
954 \ifnum\ctr@size=2
955 \@@VECTORCOPY(#1)(#2)
956 \else \@@@VECTORCOPY(#1)(#2)\fi}

\VECTORGLOBALCOPY \VECTORGLOBALCOPY is the global version of \VECTORCOPY
957 \def\@@VECTORGLOBALCOPY(#1,#2)(#3,#4){%
958 \GLOBALCOPY{#1}{#3}\GLOBALCOPY{#2}{#4}}
959
960 \def\@@@VECTORGLOBALCOPY(#1,#2,#3)(#4,#5,#6){%
961 \GLOBALCOPY{#1}{#4}\GLOBALCOPY{#2}{#5}\GLOBALCOPY{#3}{#6}}
962
963 \def\VECTORGLOBALCOPY(#1)(#2){%
964 \VECTORSIZE(#1){\ctr@size}
965 \ifnum\ctr@size=2
966 \@@VECTORGLOBALCOPY(#1)(#2)
967 \else \@@@VECTORGLOBALCOPY(#1)(#2)\fi}

\@OUTPUTVECTOR
968 \def\@@OUTPUTVECTOR(#1,#2){%
969 \VECTORGLOBALCOPY(#1,#2)(\ctr@outa,\ctr@outb)
970 \endgroup\VECTORCOPY(\ctr@outa,\ctr@outb)(#1,#2)}
971
972 \def\@@@OUTPUTVECTOR(#1,#2,#3){%
973 \VECTORGLOBALCOPY(#1,#2,#3)(\ctr@outa,\ctr@outb,\ctr@outc)
974 \endgroup\VECTORCOPY(\ctr@outa,\ctr@outb,\ctr@outc)(#1,#2,#3)}
975
976 \def\@OUTPUTVECTOR(#1){\VECTORSIZE(#1){\ctr@size}
977 \ifnum\ctr@size=2
978 \@@OUTPUTVECTOR(#1)
979 \else \@@@OUTPUTVECTOR(#1)\fi}

\SCALARPRODUCT Scalar product of two vectors.
980 \def\@@SCALARPRODUCT(#1,#2)(#3,#4)#5{%
981 \MULTIPLY{#1}{#3}{#5}
982 \MULTIPLY{#2}{#4}\ctr@tempa
983 \ADD{#5}{\ctr@tempa}{#5}}
984

```

```

985 \def\@@@SCALARPRODUCT(#1,#2,#3)(#4,#5,#6)#7{%
986     \MULTIPLY{#1}{#4}{#7}
987     \MULTIPLY{#2}{#5}\cctr@tempa
988     \ADD{#7}{\cctr@tempa}{#7}
989     \MULTIPLY{#3}{#6}\cctr@tempa
990     \ADD{#7}{\cctr@tempa}{#7}}
991
992 \def\SCALARPRODUCT(#1)(#2)#3{%
993     \begingroup
994     \VECTORSIZE(#1){\cctr@size}
995     \ifnum\cctr@size=2
996         \@@@SCALARPRODUCT(#1)(#2){#3}
997     \else \@@@SCALARPRODUCT(#1)(#2){#3}\fi\@OUTPUTSOL{#3}}

```

\DOTPRODUCT \DOTPRODUCT is an alias for \SCALARPRODUCT.

```

998 \let\DOTPRODUCT\SCALARPRODUCT

```

\VECTORPRODUCT Vector product of two (three dimensional) vectors.

```

999 \def\@@VECTORPRODUCT(#1)(#2)(#3,#4){%
1000     \let#3\undefined
1001     \let#4\undefined
1002     \cctr@Warncrossprod(#1)(#2)}
1003
1004 \def\@@@VECTORPRODUCT(#1,#2,#3)(#4,#5,#6)(#7,#8,#9){%
1005     \DETERMINANT(#2,#3;#5,#6){#7}
1006     \DETERMINANT(#3,#1;#6,#4){#8}
1007     \DETERMINANT(#1,#2;#4,#5){#9}}
1008
1009 \def\VECTORPRODUCT(#1)(#2)(#3){%
1010     \begingroup
1011     \VECTORSIZE(#1){\cctr@size}
1012     \ifnum\cctr@size=2
1013         \@@VECTORPRODUCT(#1)(#2)(#3)
1014     \else \@@@VECTORPRODUCT(#1)(#2)(#3)\fi\@OUTPUTSOL{#3}}

```

\CROSSPRODUCT \CROSSPRODUCT is an alias for \VECTORPRODUCT.

```

1015 \let\CROSSPRODUCT\VECTORPRODUCT

```

\VECTORADD Sum of two vectors.

```

1016 \def\@@VECTORADD(#1,#2)(#3,#4)(#5,#6){%
1017     \ADD{#1}{#3}{#5}
1018     \ADD{#2}{#4}{#6}}
1019
1020 \def\@@@VECTORADD(#1,#2,#3)(#4,#5,#6)(#7,#8,#9){%
1021     \ADD{#1}{#4}{#7}
1022     \ADD{#2}{#5}{#8}
1023     \ADD{#3}{#6}{#9}}
1024
1025 \def\VECTORADD(#1)(#2)(#3){%

```



```

1026     \VECTORSIZE(#1){\cctr@size}
1027     \ifnum\cctr@size=2
1028         \@@VECTORADD(#1)(#2)(#3)
1029     \else \@@@VECTORADD(#1)(#2)(#3)\fi}

```

**\VECTORSUB** Difference of two vectors.

```

1030 \def\@@VECTORSUB(#1,#2)(#3,#4)(#5,#6){%
1031     \VECTORADD(#1,#2)(-#3,-#4)(#5,#6)}
1032
1033 \def\@@@VECTORSUB(#1,#2,#3)(#4,#5,#6)(#7,#8,#9){%
1034     \VECTORADD(#1,#2,#3)(-#4,-#5,-#6)(#7,#8,#9)}
1035
1036 \def\VECTORSUB(#1)(#2)(#3){%
1037     \VECTORSIZE(#1){\cctr@size}
1038     \ifnum\cctr@size=2
1039         \@@VECTORSUB(#1)(#2)(#3)
1040     \else \@@@VECTORSUB(#1)(#2)(#3)\fi}

```

**\VECTORABSVALUE** Absolute value of a each entry of a vector.

```

1041 \def\@@VECTORABSVALUE(#1,#2)(#3,#4){%
1042     \ABSVALUE{#1}{#3}\ABSVALUE{#2}{#4}}
1043
1044 \def\@@@VECTORABSVALUE(#1,#2,#3)(#4,#5,#6){%
1045     \ABSVALUE{#1}{#4}\ABSVALUE{#2}{#5}\ABSVALUE{#3}{#6}}
1046
1047 \def\VECTORABSVALUE(#1)(#2){%
1048     \VECTORSIZE(#1){\cctr@size}
1049     \ifnum\cctr@size=2
1050         \@@VECTORABSVALUE(#1)(#2)
1051     \else \@@@VECTORABSVALUE(#1)(#2)\fi}

```

**\SCALARVECTORPRODUCT** Scalar-vector product.

```

1052 \def\@@SCALARVECTORPRODUCT#1(#2,#3)(#4,#5){%
1053     \MULTIPLY{#1}{#2}{#4}
1054     \MULTIPLY{#1}{#3}{#5}}
1055
1056 \def\@@@SCALARVECTORPRODUCT#1(#2,#3,#4)(#5,#6,#7){%
1057     \MULTIPLY{#1}{#2}{#5}
1058     \MULTIPLY{#1}{#3}{#6}
1059     \MULTIPLY{#1}{#4}{#7}}
1060
1061 \def\SCALARVECTORPRODUCT#1(#2)(#3){%
1062     \VECTORSIZE(#2){\cctr@size}
1063     \ifnum\cctr@size=2
1064         \@@SCALARVECTORPRODUCT{#1}(#2)(#3)
1065     \else \@@@SCALARVECTORPRODUCT{#1}(#2)(#3)\fi}

```

**\VECTORNORM** Euclidean norm of a vector.

```

1066 \def\VECTORNORM(#1)#2{%
1067     \begingroup

```

```

1068     \SCALARPRODUCT(#1)(#1){\cctr@temp}
1069     \SQUAREROOT{\cctr@temp}{#2}\@OUTPUTSOL{#2}}

```

**\UNITVECTOR** Unitary vector parallel to a given vector.

```

1070 \def\UNITVECTOR(#1)(#2){%
1071     \begingroup
1072     \VECTORNORM(#1){\cctr@tempa}
1073     \DIVIDE{1}{\cctr@tempa}{\cctr@tempa}
1074     \SCALARVECTORPRODUCT{\cctr@tempa}{#1)(#2)\@OUTPUTVECTOR(#2)}

```

**\TWOVECTORSANGLE** Angle between two vectors.

```

1075 \def\TWOVECTORSANGLE(#1)(#2)#3{%
1076     \begingroup
1077     \VECTORNORM(#1){\cctr@tempa}
1078     \VECTORNORM(#2){\cctr@tempb}
1079     \SCALARPRODUCT(#1)(#2){\cctr@tempc}
1080     \ifdim \cctr@tempa\p@ =\z@
1081         \let#3\undefined
1082         \cctr@Warnnoangle(#1)(#2)
1083     \else
1084         \ifdim \cctr@tempb\p@ =\z@
1085             \let#3\undefined
1086             \cctr@Warnnoangle(#1)(#2)
1087         \else
1088             \DIVIDE{\cctr@tempc}{\cctr@tempa}{\cctr@tempc}
1089             \DIVIDE{\cctr@tempc}{\cctr@tempb}{\cctr@tempc}
1090             \ARCCOS{\cctr@tempc}{#3}
1091     \fi\fi\@OUTPUTSOL{#3}}

```

### Matrix operations

Here, we need to define some internal macros to simulate commands with more than nine arguments.

**\@TDMATRIXCOPY** This command copies a  $3 \times 3$  matrix to the commands `\cctr@solAA`, `\cctr@solAB`, ..., `\cctr@solCC`.

```

1092 \def\@TDMATRIXCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1093     \COPY{#1}{\cctr@solAA}
1094     \COPY{#2}{\cctr@solAB}
1095     \COPY{#3}{\cctr@solAC}
1096     \COPY{#4}{\cctr@solBA}
1097     \COPY{#5}{\cctr@solBB}
1098     \COPY{#6}{\cctr@solBC}
1099     \COPY{#7}{\cctr@solCA}
1100     \COPY{#8}{\cctr@solCB}
1101     \COPY{#9}{\cctr@solCC}}

```

**\@TDMATRIXSOL** This command copies the commands `\cctr@solAA`, `\cctr@solAB`, ..., `\cctr@solCC` to a  $3 \times 3$  matrix. This macro is used to store the results of a matrix operation.

```

1102 \def\@TDMATRIXSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%

```

```

1103 \COPY{\cctr@solAA}{#1}
1104 \COPY{\cctr@solAB}{#2}
1105 \COPY{\cctr@solAC}{#3}
1106 \COPY{\cctr@solBA}{#4}
1107 \COPY{\cctr@solBB}{#5}
1108 \COPY{\cctr@solBC}{#6}
1109 \COPY{\cctr@solCA}{#7}
1110 \COPY{\cctr@solCB}{#8}
1111 \COPY{\cctr@solCC}{#9}}

```

`\@TDMATRIXGLOBALSOL`

```

1112 \def\@TDMATRIXGLOBALSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1113 \GLOBALCOPY{\cctr@solAA}{#1}
1114 \GLOBALCOPY{\cctr@solAB}{#2}
1115 \GLOBALCOPY{\cctr@solAC}{#3}
1116 \GLOBALCOPY{\cctr@solBA}{#4}
1117 \GLOBALCOPY{\cctr@solBB}{#5}
1118 \GLOBALCOPY{\cctr@solBC}{#6}
1119 \GLOBALCOPY{\cctr@solCA}{#7}
1120 \GLOBALCOPY{\cctr@solCB}{#8}
1121 \GLOBALCOPY{\cctr@solCC}{#9}}

```

`\@TDMATRIXNOSOL` This command undefines a  $3 \times 3$  matrix when a matrix problem has no solution.

```

1122 \def\@TDMATRIXNOSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1123 \let#1\undefined
1124 \let#2\undefined
1125 \let#3\undefined
1126 \let#4\undefined
1127 \let#5\undefined
1128 \let#6\undefined
1129 \let#7\undefined
1130 \let#8\undefined
1131 \let#9\undefined
1132 }

```

`\@TDMATRIXSOL` This command stores or undefines the solution.

```

1133 \def\@TDMATRIXSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1134 \ifx\cctr@solAA\undefined
1135 \@TDMATRIXNOSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9)%
1136 \else
1137 \@TDMATRIXSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9)\fi}

```

`\@NUMBERSOL` This command stores the scalar solution of a matrix operation.

```

1138 \def\@NUMBERSOL#1{\COPY{\cctr@sol}{#1}}

```

`\MATRIXSIZE` Size (2 or 3) of a matrix.

```

1139 \def\MATRIXSIZE(#1)#2{\expandafter\MATRIXSIZE(#1;){#2}}
1140 \def\MATRIXSIZE(#1;#2;#3;#4)#5{\ifx$#3$\COPY{2}{#5}
1141 \else\COPY{3}{#5}\fi\ignorespaces}

```

\MATRIXCOPY Store a matrix in 4 or 9 commands.

```
1142 \def\@MATRIXCOPY(#1,#2;#3,#4)(#5,#6;#7,#8){%
1143   \COPY{#1}{#5}\COPY{#2}{#6}\COPY{#3}{#7}\COPY{#4}{#8}}
1144
1145 \def\@MATRIXCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1146   \@TDMATRIXCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9)
1147   \@TDMATRIXSOL}
1148
1149 \def\MATRIXCOPY(#1)(#2){%
1150   \MATRIXSIZE(#1){\ctr@size}
1151   \ifnum\ctr@size=2
1152     \@MATRIXCOPY(#1)(#2)
1153   \else \@MATRIXCOPY(#1)(#2)\fi}
```

\MATRIXGLOBALCOPY Global version of \MATRIXCOPY.

```
1154 \def\@MATRIXGLOBALCOPY(#1,#2;#3,#4)(#5,#6;#7,#8){%
1155   \GLOBALCOPY{#1}{#5}\GLOBALCOPY{#2}{#6}\GLOBALCOPY{#3}{#7}\GLOBALCOPY{#4}{#8}}
1156
1157 \def\@MATRIXGLOBALCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1158   \@TDMATRIXCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9)
1159   \@TDMATRIXGLOBALSOL}
1160
1161 \def\MATRIXGLOBALCOPY(#1)(#2){%
1162   \MATRIXSIZE(#1){\ctr@size}
1163   \ifnum\ctr@size=2
1164     \@MATRIXGLOBALCOPY(#1)(#2)
1165   \else \@MATRIXGLOBALCOPY(#1)(#2)\fi}
```

\@OUTPUTMATRIX

```
1166 \def\@OUTPUTMATRIX(#1,#2;#3,#4){%
1167   \MATRIXGLOBALCOPY(#1,#2;#3,#4)(\ctr@outa,\ctr@outb;\ctr@outc,\ctr@outd)
1168   \endgroup\MATRIXCOPY(\ctr@outa,\ctr@outb;\ctr@outc,\ctr@outd)(#1,#2;#3,#4)}
1169
1170 \def\@OUTPUTMATRIX(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1171   \MATRIXGLOBALCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9)(%
1172     \ctr@outa,\ctr@outb,\ctr@outc;
1173     \ctr@outd,\ctr@oute,\ctr@outf;
1174     \ctr@outg,\ctr@outh,\ctr@outi)
1175   \endgroup\MATRIXCOPY(%
1176     \ctr@outa,\ctr@outb,\ctr@outc;
1177     \ctr@outd,\ctr@oute,\ctr@outf;
1178     \ctr@outg,\ctr@outh,\ctr@outi)(#1,#2,#3;#4,#5,#6;#7,#8,#9)}
1179
1180 \def\@OUTPUTMATRIX(#1){\MATRIXSIZE(#1){\ctr@size}
1181   \ifnum\ctr@size=2
1182     \@OUTPUTMATRIX(#1)
1183   \else \@OUTPUTMATRIX(#1)\fi}
```

\TRANPOSEMATRIX Matrix transposition.

```
1184 \def\@TRANPOSEMATRIX(#1,#2;#3,#4)(#5,#6;#7,#8){%
```

```

1185     \COPY{#1}{#5}\COPY{#3}{#6}\COPY{#2}{#7}\COPY{#4}{#8}}
1186
1187 \def\@@@TRANPOSEMATRIX(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1188     \@TDMATRIXCOPY(#1,#4,#7;#2,#5,#8;#3,#6,#9)
1189     \@TDMATRIXSOL}
1190
1191 \def\TRANPOSEMATRIX(#1)(#2){%
1192     \begingroup
1193     \MATRIXSIZE(#1){\ctr@size}
1194     \ifnum\ctr@size=2
1195         \@@TRANPOSEMATRIX(#1)(#2)
1196     \else \@@@TRANPOSEMATRIX(#1)(#2)\fi\@OUTPUTMATRIX(#2)}

```

\MATRIXADD Sum of two matrices.

```

1197 \def\@@MATRIXADD(#1;#2)(#3;#4)(#5,#6;#7,#8){%
1198     \VECTORADD(#1)(#3)(#5,#6)
1199     \VECTORADD(#2)(#4)(#7,#8)}
1200
1201 \def\@@@MATRIXADD(#1;#2;#3)(#4;#5;#6){%
1202     \VECTORADD(#1)(#4)(\ctr@solAA,\ctr@solAB,\ctr@solAC)
1203     \VECTORADD(#2)(#5)(\ctr@solBA,\ctr@solBB,\ctr@solBC)
1204     \VECTORADD(#3)(#6)(\ctr@solCA,\ctr@solCB,\ctr@solCC)
1205     \@TDMATRIXSOL}
1206
1207 \def\MATRIXADD(#1)(#2)(#3){%
1208     \begingroup
1209     \MATRIXSIZE(#1){\ctr@size}
1210     \ifnum\ctr@size=2
1211         \@@MATRIXADD(#1)(#2)(#3)
1212     \else \@@@MATRIXADD(#1)(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```

\MATRIXSUB Difference of two matrices.

```

1213 \def\@@MATRIXSUB(#1;#2)(#3;#4)(#5,#6;#7,#8){%
1214     \VECTORSUB(#1)(#3)(#5,#6)
1215     \VECTORSUB(#2)(#4)(#7,#8)}
1216
1217 \def\@@@MATRIXSUB(#1;#2;#3)(#4;#5;#6){%
1218     \VECTORSUB(#1)(#4)(\ctr@solAA,\ctr@solAB,\ctr@solAC)
1219     \VECTORSUB(#2)(#5)(\ctr@solBA,\ctr@solBB,\ctr@solBC)
1220     \VECTORSUB(#3)(#6)(\ctr@solCA,\ctr@solCB,\ctr@solCC)
1221     \@TDMATRIXSOL}
1222
1223 \def\MATRIXSUB(#1)(#2)(#3){%
1224     \begingroup
1225     \MATRIXSIZE(#1){\ctr@size}
1226     \ifnum\ctr@size=2
1227         \@@MATRIXSUB(#1)(#2)(#3)
1228     \else \@@@MATRIXSUB(#1)(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```

\MATRIXABSVALUE Absolute value (of each entry) of a matrix.

```

1229 \def\@@MATRIXABSVALUE(#1;#2)(#3;#4){%
1230     \VECTORABSVALUE(#1)(#3)\VECTORABSVALUE(#2)(#4)}
1231
1232 \def\@@@MATRIXABSVALUE(#1;#2;#3)(#4;#5;#6){%
1233     \VECTORABSVALUE(#1)(#4)\VECTORABSVALUE(#2)(#5)\VECTORABSVALUE(#3)(#6)}
1234
1235 \def\MATRIXABSVALUE(#1)(#2){%
1236     \begingroup
1237     \MATRIXSIZE(#1){\cctr@size}
1238     \ifnum\cctr@size=2
1239         \@@MATRIXABSVALUE(#1)(#2)
1240     \else \@@@MATRIXABSVALUE(#1)(#2)\fi\@OUTPUTMATRIX(#2)}

```

\MATRIXVECTORPRODUCT Matrix-vector product.

```

1241 \def\@@MATRIXVECTORPRODUCT(#1;#2)(#3)(#4,#5){%
1242     \SCALARPRODUCT(#1)(#3){#4}
1243     \SCALARPRODUCT(#2)(#3){#5}}
1244
1245 \def\@@@MATRIXVECTORPRODUCT(#1;#2;#3)(#4)(#5,#6,#7){%
1246     \SCALARPRODUCT(#1)(#4){#5}
1247     \SCALARPRODUCT(#2)(#4){#6}
1248     \SCALARPRODUCT(#3)(#4){#7}}
1249
1250 \def\MATRIXVECTORPRODUCT(#1)(#2)(#3){%
1251     \begingroup
1252     \MATRIXSIZE(#1){\cctr@size}
1253     \ifnum\cctr@size=2
1254         \@@MATRIXVECTORPRODUCT(#1)(#2)(#3)
1255     \else \@@@MATRIXVECTORPRODUCT(#1)(#2)(#3)\fi\@OUTPUTVECTOR(#3)}

```

\VECTORMATRIXPRODUCT Vector-matrix product.

```

1256 \def\@@VECTORMATRIXPRODUCT(#1)(#2,#3;#4,#5)(#6,#7){%
1257     \SCALARPRODUCT(#1)(#2,#4){#6}
1258     \SCALARPRODUCT(#1)(#3,#5){#7}}
1259
1260 \def\@@@VECTORMATRIXPRODUCT(#1,#2,#3)(#4;#5;#6)(#7){%
1261     \SCALARVECTORPRODUCT{#1}{#4)(#7)
1262     \SCALARVECTORPRODUCT{#2}{#5)(\cctr@tempa,\cctr@tempb,\cctr@tempc)
1263     \VECTORADD(#7)(\cctr@tempa,\cctr@tempb,\cctr@tempc)(#7)
1264     \SCALARVECTORPRODUCT{#3}{#6)(\cctr@tempa,\cctr@tempb,\cctr@tempc)
1265     \VECTORADD(#7)(\cctr@tempa,\cctr@tempb,\cctr@tempc)(#7)}
1266
1267 \def\VECTORMATRIXPRODUCT(#1)(#2)(#3){%
1268     \begingroup
1269     \VECTORSIZE(#1){\cctr@size}
1270     \ifnum\cctr@size=2
1271         \@@VECTORMATRIXPRODUCT(#1)(#2)(#3)
1272     \else \@@@VECTORMATRIXPRODUCT(#1)(#2)(#3)\fi\@OUTPUTVECTOR(#3)}

```

\SCALARMATRIXPRODUCT Scalar-matrix product.

```

1273 \def\@@SCALARMATRIXPRODUCT#1(#2;#3)(#4,#5;#6,#7){%
1274     \SCALARVECTORPRODUCT{#1}(#2)(#4,#5)
1275     \SCALARVECTORPRODUCT{#1}(#3)(#6,#7)}
1276
1277 \def\@@@SCALARMATRIXPRODUCT#1(#2;#3;#4){%
1278     \SCALARVECTORPRODUCT{#1}(#2)(\cctr@solAA,\cctr@solAB,\cctr@solAC)
1279     \SCALARVECTORPRODUCT{#1}(#3)(\cctr@solBA,\cctr@solBB,\cctr@solBC)
1280     \SCALARVECTORPRODUCT{#1}(#4)(\cctr@solCA,\cctr@solCB,\cctr@solCC)
1281     \@TDMATRIXSOL}
1282
1283 \def\SCALARMATRIXPRODUCT#1(#2)(#3){%
1284     \begingroup
1285     \MATRIXSIZE(#2){\cctr@size}
1286     \ifnum\cctr@size=2
1287         \@@SCALARMATRIXPRODUCT{#1}(#2)(#3)
1288     \else \@@@SCALARMATRIXPRODUCT{#1}(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```

\MATRIXPRODUCT Product of two matrices.

```

1289 \def\@@MATRIXPRODUCT(#1)(#2,#3;#4,#5)(#6,#7;#8,#9){%
1290     \MATRIXVECTORPRODUCT(#1)(#2,#4)(#6,#8)
1291     \MATRIXVECTORPRODUCT(#1)(#3,#5)(#7,#9)}
1292
1293 \def\@@@MATRIXPRODUCT(#1;#2;#3)(#4){%
1294     \VECTORMATRIXPRODUCT(#1)(#4)(\cctr@solAA,\cctr@solAB,\cctr@solAC)
1295     \VECTORMATRIXPRODUCT(#2)(#4)(\cctr@solBA,\cctr@solBB,\cctr@solBC)
1296     \VECTORMATRIXPRODUCT(#3)(#4)(\cctr@solCA,\cctr@solCB,\cctr@solCC)
1297     \@TDMATRIXSOL}
1298
1299 \def\MATRIXPRODUCT(#1)(#2)(#3){%
1300     \begingroup
1301     \MATRIXSIZE(#1){\cctr@size}
1302     \ifnum\cctr@size=2
1303         \@@MATRIXPRODUCT(#1)(#2)(#3)
1304     \else \@@@MATRIXPRODUCT(#1)(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```

\DETERMINANT Determinant of a matrix.

```

1305 \def\@@DETERMINANT(#1,#2;#3,#4)#5{%
1306     \MULTIPLY{#1}{#4}{#5}
1307     \MULTIPLY{#2}{#3}{\cctr@tempa}
1308     \SUBTRACT{#5}{\cctr@tempa}{#5}}
1309
1310 \def\@@@DETERMINANT(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1311     \DETERMINANT(#5,#6;#8,#9){\cctr@det}\MULTIPLY{#1}{\cctr@det}{\cctr@sol}
1312     \DETERMINANT(#6,#4;#9,#7){\cctr@det}\MULTIPLY{#2}{\cctr@det}{\cctr@det}
1313         \ADD{\cctr@sol}{\cctr@det}{\cctr@sol}
1314     \DETERMINANT(#4,#5;#7,#8){\cctr@det}\MULTIPLY{#3}{\cctr@det}{\cctr@det}
1315         \ADD{\cctr@sol}{\cctr@det}{\cctr@sol}
1316     \@NUMBERSOL}
1317
1318 \def\DETERMINANT(#1)#2{%

```

```

1319     \begingroup
1320     \MATRIXSIZE(#1){\cctr@size}
1321     \ifnum\cctr@size=2
1322         \@DETERMINANT(#1){#2}
1323     \else \@DETERMINANT(#1){#2}\fi\@OUTPUTSOL{#2}}

```

\INVERSEMATRIX Inverse of a matrix.

```

1324 \def\@INVERSEMATRIX(#1,#2;#3,#4)(#5,#6;#7,#8){%
1325     \ifdim \cctr@det\p@ <\cctr@epsilon % Matrix is singular
1326         \let#5\undefined
1327         \let#6\undefined
1328         \let#7\undefined
1329         \let#8\undefined
1330         \cctr@Warnsingmatrix{#1}{#2}{#3}{#4}%
1331     \else \COPY{#1}{#8}
1332         \COPY{#4}{#5}
1333         \MULTIPLY{-1}{#3}{#7}
1334         \MULTIPLY{-1}{#2}{#6}
1335         \DIVIDE{1}{\cctr@det}{\cctr@det}
1336         \SCALARMATRIXPRODUCT{\cctr@det}{#5,#6;#7,#8)(#5,#6;#7,#8)
1337     \fi}
1338
1339 \def\@@INVERSEMATRIX(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1340     \ifdim \cctr@det\p@ <\cctr@epsilon % Matrix is singular
1341         \@TDMATRIXNOSOL(\cctr@solAA,\cctr@solAB,\cctr@solAC;
1342             \cctr@solBA,\cctr@solBB,\cctr@solBC;
1343             \cctr@solCA,\cctr@solCB,\cctr@solCC)
1344         \cctr@WarnsingTDMatrix{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}{#9}%
1345     \else
1346         \@ADJMATRIX(#1,#2,#3;#4,#5,#6;#7,#8,#9)
1347         \@SCLRDIIVECT{\cctr@det}(\cctr@solAA,\cctr@solAB,\cctr@solAC)(%
1348             \cctr@solAA,\cctr@solAB,\cctr@solAC)
1349         \@SCLRDIIVECT{\cctr@det}(\cctr@solBA,\cctr@solBB,\cctr@solBC)(%
1350             \cctr@solBA,\cctr@solBB,\cctr@solBC)
1351         \@SCLRDIIVECT{\cctr@det}(\cctr@solCA,\cctr@solCB,\cctr@solCC)(%
1352             \cctr@solCA,\cctr@solCB,\cctr@solCC)
1353     \fi
1354     \@TDMATRIXSOL}
1355
1356 \def\@SCLRDIIVECT#1(#2,#3,#4)(#5,#6,#7){%
1357     \DIVIDE{#2}{#1}{#5}\DIVIDE{#3}{#1}{#6}\DIVIDE{#4}{#1}{#7}}
1358
1359 \def\@ADJMATRIX(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1360     \DETERMINANT(#5,#6;#8,#9){\cctr@solAA}
1361     \DETERMINANT(#6,#4;#9,#7){\cctr@solBA}
1362     \DETERMINANT(#4,#5;#7,#8){\cctr@solCA}
1363     \DETERMINANT(#8,#9;#2,#3){\cctr@solAB}
1364     \DETERMINANT(#1,#3;#7,#9){\cctr@solBB}
1365     \DETERMINANT(#2,#1;#8,#7){\cctr@solCB}
1366     \DETERMINANT(#2,#3;#5,#6){\cctr@solAC}

```



```

1367         \DETERMINANT(#3,#1,#6,#4){\cctr@solBC}
1368         \DETERMINANT(#1,#2,#4,#5){\cctr@solCC}}
1369
1370 \def\INVERSEMATRIX(#1)(#2){%
1371     \begingroup
1372     \DETERMINANT(#1){\cctr@det}
1373     \ABSVALUE{\cctr@det}{\cctr@det}
1374     \MATRIXSIZE(#1){\cctr@size}
1375     \ifnum\cctr@size=2
1376         \@INVERSEMATRIX(#1)(#2)
1377     \else
1378         \@@INVERSEMATRIX(#1)(#2)\fi\@OUTPUTMATRIX(#2)}

```

\SOLVELINEARSYSTEM Solving a linear system (two equations and two unknowns or three equations and three unknowns).

```

1379 \def\@INCSYS#1#2{\cctr@WarnIncLinSys
1380     \let#1\undefined\let#2\undefined}
1381
1382 \def\@SOLPART#1#2#3#4{\cctr@WarnIndLinSys
1383     \DIVIDE{#1}{#2}{#3}
1384     \COPY{0}{#4}}
1385
1386 \def\@TDINCSYS(#1,#2,#3){\cctr@WarnIncTDLinSys
1387     \let#1\undefined
1388     \let#2\undefined
1389     \let#3\undefined}
1390
1391 \def\@@SOLVELINEARSYSTEM(#1,#2;#3,#4)(#5,#6)(#7,#8){%
1392     \DETERMINANT(#1,#2;#3,#4)\cctr@deta
1393     \DETERMINANT(#5,#2;#6,#4)\cctr@detb
1394     \DETERMINANT(#1,#5;#3,#6)\cctr@detc
1395     \ABSVALUE{\cctr@deta}{\cctr@deta}
1396     \ABSVALUE{\cctr@detb}{\cctr@detb}
1397     \ABSVALUE{\cctr@detc}{\cctr@detc}
1398     \ifdim \cctr@deta\p@>\cctr@epsilon% Regular matrix. Determinate system
1399         \DIVIDE{\cctr@detb}{\cctr@deta}{#7}
1400         \DIVIDE{\cctr@detc}{\cctr@deta}{#8}
1401     \else % Singular matrix \cctr@deta=0
1402         \ifdim \cctr@deta\p@>\cctr@epsilon% Incompatible system
1403             \@INCSYS#7#8
1404         \else
1405             \ifdim \cctr@detc\p@>\cctr@epsilon% Incompatible system
1406                 \@INCSYS#7#8
1407             \else
1408                 \MATRIXABSVALUE(#1,#2;#3,#4)(\cctr@tempa,\cctr@tempb;
1409                     \cctr@tempc,\cctr@tempd)
1410                 \ifdim \cctr@tempa\p@ > \cctr@epsilon
1411                     % Indeterminate system
1412                 \@SOLPART{#5}{#1}{#7}{#8}
1413             \else

```

```

1414         \ifdim \cctr@tempb\p@ > \cctr@epsilon
1415             % Indeterminate system
1416         \@SOLPART{#5}{#2}{#8}{#7}
1417     \else
1418         \ifdim \cctr@tempc\p@ > \cctr@epsilon
1419             % Indeterminate system
1420         \@SOLPART{#6}{#3}{#7}{#8}
1421     \else
1422         \ifdim \cctr@tempd\p@ > \cctr@epsilon
1423             % Indeterminate system
1424         \@SOLPART{#6}{#4}{#8}{#7}
1425     \else
1426         \VECTORNORM(#5,#6){\cctr@tempa}
1427         \ifdim \cctr@tempa\p@ > \cctr@epsilon
1428             % Incompatible system
1429         \@INCSYS#7#8
1430     \else
1431         \cctr@WarnZeroLinSys
1432         \COPY{0}{#7}\COPY{0}{#8}
1433         % 0x=0 Indeterminate system
1434     \fi\fi\fi\fi\fi\fi\fi\fi}
1435
1436 \def\@@@SOLVELINEARSYSTEM(#1)(#2)(#3){%
1437     \DETERMINANT(#1){\cctr@det}
1438     \ABSVALUE{\cctr@det}{\cctr@@det}
1439     \ifdim\cctr@@det\p@<\cctr@epsilon
1440         \@TDINCSYS(#3)
1441     \else
1442         \@ADJMATRIX(#1)
1443         \MATRIXVECTORPRODUCT(\cctr@solAA,\cctr@solAB,\cctr@solAC;
1444             \cctr@solBA,\cctr@solBB,\cctr@solBC;
1445             \cctr@solCA,\cctr@solCB,\cctr@solCC)(#2)(#3)
1446         \@SCLRDIVVECT{\cctr@det}{#3)(#3)
1447     \fi}
1448
1449 \def\SOLVELINEARSYSTEM(#1)(#2)(#3){%
1450     \begingroup
1451     \MATRIXSIZE(#1){\cctr@size}
1452     \ifnum\cctr@size=2
1453         \@@@SOLVELINEARSYSTEM(#1)(#2)(#3)
1454     \else
1455         \@@@SOLVELINEARSYSTEM(#1)(#2)(#3)
1456     \fi\@OUTPUTVECTOR(#3)}

```

## Predefined numbers

\numberPI The number  $\pi$

```
1457 \def\numberPI{3.14159}
```

\numberTWOPI  $2\pi$

$\backslash\text{numberPI}\backslash\text{numberTWOPI}$  1458  $\backslash\text{MULTIPLY}\{\backslash\text{numberPI}\}\{2\}\{\backslash\text{numberTWOPI}\}$   
 $\backslash\text{numberHALFPI}$   $\pi/2$   
 $\backslash\text{numberHALFPI}$  1459  $\backslash\text{DIVIDE}\{\backslash\text{numberPI}\}\{2\}\{\backslash\text{numberHALFPI}\}$   
 $\backslash\text{numberTHREEHALFPI}$   $3\pi/2$   
 $\backslash\text{numberTHREEHALFPI}$  1460  $\backslash\text{MULTIPLY}\{\backslash\text{numberPI}\}\{1.5\}\{\backslash\text{numberTHREEHALFPI}\}$   
 $\backslash\text{numberTHIRDPI}$   $\pi/3$   
 $\backslash\text{numberTHIRDPI}$  1461  $\backslash\text{DIVIDE}\{\backslash\text{numberPI}\}\{3\}\{\backslash\text{numberTHIRDPI}\}$   
 $\backslash\text{numberQUARTERPI}$   $\pi/4$   
 $\backslash\text{numberQUARTERPI}$  1462  $\backslash\text{DIVIDE}\{\backslash\text{numberPI}\}\{4\}\{\backslash\text{numberQUARTERPI}\}$   
 $\backslash\text{numberFIFTHPI}$   $\pi/5$   
 $\backslash\text{numberFIFTHPI}$  1463  $\backslash\text{DIVIDE}\{\backslash\text{numberPI}\}\{5\}\{\backslash\text{numberFIFTHPI}\}$   
 $\backslash\text{numberSIXTHPI}$   $\pi/6$   
 $\backslash\text{numberSIXTHPI}$  1464  $\backslash\text{DIVIDE}\{\backslash\text{numberPI}\}\{6\}\{\backslash\text{numberSIXTHPI}\}$   
 $\backslash\text{numberE}$  The number e  
 $\backslash\text{numberE}$  1465  $\backslash\text{def}\backslash\text{numberE}\{2.71828\}$   
 $\backslash\text{numberINVE}$   $1/e$   
 $\backslash\text{numberINVE}$  1466  $\backslash\text{DIVIDE}\{1\}\{\backslash\text{numberE}\}\{\backslash\text{numberINVE}\}$   
 $\backslash\text{numberETWO}$   $e^2$   
 $\backslash\text{numberETWO}$  1467  $\backslash\text{SQUARE}\{\backslash\text{numberE}\}\{\backslash\text{numberETWO}\}$   
 $\backslash\text{numberINVETWO}$   $1/e^2$   
 $\backslash\text{numberINVETWO}$  1468  $\backslash\text{SQUARE}\{\backslash\text{numberINVE}\}\{\backslash\text{numberINVETWO}\}$   
 $\backslash\text{numberLOGTEN}$   $\log 10$   
 $\backslash\text{numberLOGTEN}$  1469  $\backslash\text{def}\backslash\text{numberLOGTEN}\{2.30258\}$   
 $\backslash\text{numberGOLD}$  The golden ratio  $\phi$   
 $\backslash\text{numberGOLD}$  1470  $\backslash\text{def}\backslash\text{numberGOLD}\{1.61803\}$   
 $\backslash\text{numberINVGOLD}$   $1/\phi$   
 $\backslash\text{numberINVGOLD}$  1471  $\backslash\text{def}\backslash\text{numberINVGOLD}\{0.61803\}$   
 $\backslash\text{numberSQRTTWO}$   $\sqrt{2}$   
 $\backslash\text{numberSQRTTWO}$  1472  $\backslash\text{def}\backslash\text{numberSQRTTWO}\{1.41421\}$   
 $\backslash\text{numberSQRTHREE}$   $\sqrt{3}$   
 $\backslash\text{numberSQRTHREE}$  1473  $\backslash\text{def}\backslash\text{numberSQRTHREE}\{1.73205\}$   
 $\backslash\text{numberSQRTFIVE}$   $\sqrt{5}$   
 $\backslash\text{numberSQRTFIVE}$  1474  $\backslash\text{def}\backslash\text{numberSQRTFIVE}\{2.23607\}$

```

\numberCOSXLV  cos 45° (or cos  $\pi/4$ )
1475 \def\numberCOSXLV{0.70711}

\numberCOSXXX  cos 30° (or cos  $\pi/6$ )
1476 \def\numberCOSXXX{0.86603}

1477 \calculator

```

## 14 calculus

```

1478 (*calculus)
1479 \NeedsTeXFormat{LaTeX2e}
1480 \ProvidesPackage{calculus}[2014/02/20 v.2.0]

```

This package requires the calculator package.

```
1481 \RequirePackage{calculator}
```

### 14.1 Error and info messages

#### For scalar functions

Error message to be issued when you attempt to define, with `\newfunction`, an already defined command:

```

1482 \def\ccls@ErrorFuncDef#1{%
1483     \PackageError{calculus}%
1484     {\noexpand#1 command already defined}
1485     {The \noexpand#1 control sequence is already defined\MessageBreak
1486     If you want to redefine the \noexpand#1 command as a
1487     function\MessageBreak
1488     please, use the \noexpand\renewfunction command}}

```

Error message to be issued when you attempt to redefine, with `\renewfunction`, an undefined command:

```

1489 \def\ccls@ErrorFuncUnDef#1{%
1490     \PackageError{calculus}%
1491     {\noexpand#1 command undefined}
1492     {The \noexpand#1 control sequence is not currently defined\MessageBreak
1493     If you want to define the \noexpand#1 command as a function\MessageBreak
1494     please, use the \noexpand\newfunction command}}

```

Info message to be issued when `\ensurefunction` does not changes an already defined command:

```

1495 \def\ccls@InfoFuncEns#1{%
1496     \PackageInfo{calculus}%
1497     {\noexpand#1 command already defined\MessageBreak
1498     the \noexpand\ensurefunction command will not redefine it}}

```

#### For polar functions

```

1499 \def\ccls@ErrorPFuncDef#1{%
1500     \PackageError{calculus}%
1501     {\noexpand#1 command already defined}

```

```

1502      {The \noexpand#1 control sequence is already defined\MessageBreak
1503      If you want to redefine the \noexpand#1
1504      command as a polar function\MessageBreak
1505      please, use the \noexpand\renewpolarfunction command}}
1506
1507 \def\ccls@ErrorPFuncUnDef#1{%
1508   \PackageError{calculus}%
1509   {\noexpand#1 command undefined}
1510   {The \noexpand#1 control sequence
1511   is not currently defined.\MessageBreak
1512   If you want to define the \noexpand#1 command as a polar
1513   function\MessageBreak
1514   please, use the \noexpand\newpolarfunction command}}
1515
1516 \def\ccls@InfoPFuncEns#1{%
1517   \PackageInfo{calculus}%
1518   {\noexpand#1 command already defined\MessageBreak
1519   the \noexpand\ensurepolarfunction command does not redefine it}}

```

### For vector functions

```

1520 \def\ccls@ErrorVFuncDef#1{%
1521   \PackageError{calculus}%
1522   {\noexpand#1 command already defined}
1523   {The \noexpand#1 control sequence is already defined\MessageBreak
1524   If you want to redefine the \noexpand#1 command as a vector
1525   function\MessageBreak
1526   please, use the \noexpand\renewvectorfunction command}}
1527
1528 \def\ccls@ErrorVFuncUnDef#1{%
1529   \PackageError{calculus}%
1530   {\noexpand#1 command undefined}
1531   {The \noexpand#1 control sequence is not currently
1532   defined.\MessageBreak
1533   If you want to define the \noexpand#1 command as a vector
1534   function\MessageBreak
1535   please, use the \noexpand\newvectorfunction command}}
1536
1537 \def\ccls@InfoVFuncEns#1{%
1538   \PackageInfo{calculus}%
1539   {\noexpand#1 command already defined\MessageBreak
1540   the \noexpand\ensurevectorfunction command does not redefine it}}

```

## 14.2 New functions

### New scalar functions

`\newfunction` The `\newfunction{#1}{#2}` instruction defines a new function called #1. #2 is the list of instructions to calculate the function  $\backslash y$  and his derivative  $\backslash Dy$  from the  $\backslash t$  variable.

```

1541 \def\newfunction#1#2{%
1542   \ifx #1\undefined
1543     \ccls@deffunction{#1}{#2}

```

```

1544     \else
1545     \ccls@ErrorFuncDef{#1}
1546     \fi}

```

`\renewfunction` `\renewfunction` redefines #1, as a new function, if this command is already defined.

```

1547 \def\renewfunction#1#2{%
1548     \ifx #1\undefined
1549     \ccls@ErrorFuncUnDef{#1}
1550     \else
1551     \ccls@deffunction{#1}{#2}
1552     \fi}

```

`\ensurefunction` `\ensurefunction` defines the new function #1 (only if this macro is undefined).

```

1553 \def\ensurefunction#1#2{%
1554     \ifx #1\undefined\ccls@deffunction{#1}{#2}
1555     \else
1556     \ccls@InfoFuncEns{#1}
1557     \fi}

```

`\forcefunction` `\forcefunction` defines (if undefined) or redefines (if defined) the new function #1.

```

1558 \def\forcefunction#1#2{%
1559     \ccls@deffunction{#1}{#2}}

```

`\ccls@deffunction` The private `\ccls@deffunction` command makes the real work. The new functions will have three arguments: ##1, a number, ##2, the value of the new function in that number, and ##3, the derivative.

```

1560 \def\ccls@deffunction#1#2{%
1561     \def#1##1##2##3{%
1562         \begingroup
1563         \def\t{##1}%
1564         #2
1565         \xdef##2{\y}%
1566         \xdef##3{\Dy}%
1567         \endgroup}\ignorespaces}

```

### New polar functions

`\newpolarfunction` The `\newpolarfunction{#1}{#2}` instruction defines a new polar function called #1. #2 is the list of instructions to calculate the radius  $r$  and his derivative  $Dr$  from the  $t$  arc variable.

```

1568 \def\newpolarfunction#1#2{%
1569     \ifx #1\undefined
1570     \ccls@defpolarfunction{#1}{#2}
1571     \else
1572     \ccls@ErrorPFuncDef{#1}
1573     \fi}

```

`\renewpolarfunction` `\renewpolarfunction` redefines #1 if already defined.

```

1574 \def\renewpolarfunction#1#2{%
1575     \ifx #1\undefined

```

```

1576         \ccls@ErrorPFuncUnDef{#1}
1577     \else
1578         \ccls@defpolarfunction{#1}{#2}
1579     \fi}

```

`\ensurepolarfunction` `\ensurepolarfunction` defines (only if undefined) #1.

```

1580 \def\ensurepolarfunction#1#2{%
1581     \ifx #1\undefined\ccls@defpolarfunction{#1}{#2}
1582     \else
1583         \ccls@InfoPFuncEns{#1}
1584     \fi}

```

`\forcepolarfunction` `\forcepolarfunction` defines (if undefined) or redefines (if defined) #1.

```

1585 \def\forcepolarfunction#1#2{%
1586     \ccls@defpolarfunction{#1}{#2}}

```

`\ccls@defpolarfunction` The private `\ccls@defpolarfunction` command makes the real work. The new functions will have three arguments: ##1, a number (the polar radius), ##2, ##3, ##4, and ##5, the x and y component functions and its derivatives at ##1.

```

1587 \def\ccls@defpolarfunction#1#2{%
1588     \def###1##2##3##4##5{%
1589         \begingroup
1590             \def\t{##1}
1591             #2
1592             \COS{\t}\ccls@cost
1593             \MULTIPLY\r\ccls@cost{\x}
1594             \SIN{\t}\ccls@sint
1595             \MULTIPLY\r\ccls@sint{\y}
1596             \MULTIPLY\ccls@cost\Dx\Dx
1597             \SUBTRACT{\Dx}{\y}{\Dx}
1598             \MULTIPLY\ccls@sint\Dr\Dy
1599             \ADD{\Dy}{\x}{\Dy}
1600             \xdef##2{\x}
1601             \xdef##3{\Dx}
1602             \xdef##4{\y}
1603             \xdef##5{\Dy}
1604         \endgroup}\ignorespaces}

```

### New vector functions

`\newvectorfunction` The `\newvectorfunction{#1}{#2}` instruction defines a new vector (parametric) function called #1. #2 is the list of instructions to calculate `\x`, `\y`, `\Dx` and `\Dy` from the `\t` arc variable.

```

1605 \def\newvectorfunction#1#2{%
1606     \ifx #1\undefined
1607         \ccls@defvectorfunction{#1}{#2}
1608     \else
1609         \ccls@ErrorVFuncDef{#1}
1610     \fi}

```

`\renewvectorfunction` `\renewvectorfunction` redefines #1 if already defined.

```
1611 \def\renewvectorfunction#1#2{%
1612     \ifx #1\undefined
1613         \ccls@ErrorVFuncUnDef{#1}
1614     \else
1615         \ccls@defvectorfunction{#1}{#2}
1616     \fi}
```

`\ensurevectorfunction` `\ensurevectorfunction` defines (only if undefined) #1.

```
1617 \def\ensurevectorfunction#1#2{%
1618     \ifx #1\undefined\ccls@defvectorfunction{#1}{#2}
1619     \else
1620         \ccls@InfoVFuncEns{#1}
1621     \fi}
```

`\forcevectorfunction` `\forcevectorfunction` defines (if undefined) or redefines (if defined) #1.

```
1622 \def\forcevectorfunction#1#2{%
1623     \ccls@defvectorfunction{#1}{#2}}
```

`\ccls@defvectorfunction` The private `\ccls@defvectorfunction` command makes the real work. The new functions will have three arguments: #1, a number, #2, #3, #4, and #5, the x and y component functions and its derivatives at #1.

```
1624 \def\ccls@defvectorfunction#1#2{%
1625     \def#1##1##2##3##4##5{%
1626         \begingroup
1627             \def\t{##1}
1628             #2
1629             \xdef##2{\x}
1630             \xdef##3{\Dx}
1631             \xdef##4{\y}
1632             \xdef##5{\Dy}
1633         \endgroup\ignorespaces}
```

## 14.3 Polynomials

### Linear (first degree) polynomials

`\newlpoly` The `\newlpoly{#1}{#2}{#3}` instruction defines the linear polynomial  $\#1 = \#2 + \#3t$ .

```
1634 \def\newlpoly#1#2#3{%
1635     \newfunction{#1}{%
1636         \ccls@lpoly{#2}{#3}}
```

`\renewlpoly` We define also the `\renewlpoly`, `\ensurelpoly` and `\forcelpoly` variants.

```
1637 \def\renewlpoly#1#2#3{%
1638     \renewfunction{#1}{%
1639         \ccls@lpoly{#2}{#3}}
```



`\ensurelpoly`

```
1640 \def\ensurelpoly#1#2#3{%
1641     \ensurefunction{#1}{%
1642         \ccls@lpoly{#2}{#3}}}
```

`\forcelpoly`

```
1643 \def\forcelpoly#1#2#3{%
1644     \forcefunction{#1}{%
1645         \ccls@lpoly{#2}{#3}}}
```

`\ccls@lpoly` The `\ccls@lpoly{#1}{#2}` macro defines the new polynomial function.

```
1646 \def\ccls@lpoly#1#2{%
1647     \MULTIPLY{#2}{\t}{\y}
1648     \ADD{\y}{#1}{\y}
1649     \COPY{#2}{\Dy}}
```

### Quadratic polynomials

`\newqpoly` The `\newqpoly{#1}{#2}{#3}{#4}` instruction defines the quadratic polynomial  $#1 = #2 + #3t + #4t^2$ .

```
1650 \def\newqpoly#1#2#3#4{%
1651     \newfunction{#1}{%
1652         \ccls@qpoly{#2}{#3}{#4}}}
```

`\renewqpoly`

```
1653 \def\renewqpoly#1#2#3#4{%
1654     \renewfunction{#1}{%
1655         \ccls@qpoly{#2}{#3}{#4}}}
```

`\ensureqpoly`

```
1656 \def\ensureqpoly#1#2#3#4{%
1657     \ensurefunction{#1}{%
1658         \ccls@qpoly{#2}{#3}{#4}}}
```

`\forceqpoly`

```
1659 \def\forceqpoly#1#2#3#4{%
1660     \forcefunction{#1}{%
1661         \ccls@qpoly{#2}{#3}{#4}}}
```

`\ccls@qpoly` The `\ccls@qpoly{#1}{#2}` macro defines the new polynomial function.

```
1662 \def\ccls@qpoly#1#2#3{%
1663     \MULTIPLY{\t}{#3}{\y}
1664     \MULTIPLY{2}{\y}{\Dy}
1665     \ADD{#2}{\Dy}{\Dy}
1666     \ADD{#2}{\y}{\y}
1667     \MULTIPLY{\t}{\y}{\y}
1668     \ADD{#1}{\y}{\y}}
```

## Cubic polynomials

`\newcpoly` The `\newcpoly{#1}{#2}{#3}{#4}{#5}` instruction defines the cubic polynomial  
$$\#1 = \#2 + \#3t + \#4t^2 + \#5t^3.$$

```
1669 \def\newcpoly#1#2#3#4#5{%
1670   \newfunction{#1}{%
1671     \ccls@cpoly{#2}{#3}{#4}{#5}}
```

`\renewcpoly`

```
1672 \def\renewcpoly#1#2#3#4#5{%
1673   \renewfunction{#1}{%
1674     \ccls@cpoly{#2}{#3}{#4}{#5}}
```

`\ensurecpoly`

```
1675 \def\ensurecpoly#1#2#3#4#5{%
1676   \ensurefunction{#1}{%
1677     \ccls@cpoly{#2}{#3}{#4}{#5}}
```

`\forcecpoly`

```
1678 \def\forcecpoly#1#2#3#4#5{%
1679   \forcefunction{#1}{%
1680     \ccls@cpoly{#2}{#3}{#4}{#5}}
```

`\ccls@cpoly` The `\ccls@cpoly{#1}{#2}` macro defines the new polynomial function.

```
1681 \def\ccls@cpoly#1#2#3#4{%
1682   \MULTIPLY{\t}{#4}{\y}
1683   \MULTIPLY{3}{\y}{\Dy}
1684   \ADD{#3}{\y}{\y}
1685   \MULTIPLY{2}{#3}{\ccls@temp}
1686   \ADD{\ccls@temp}{\Dy}{\Dy}
1687   \MULTIPLY{\t}{\y}{\y}
1688   \MULTIPLY{\t}{\Dy}{\Dy}
1689   \ADD{#2}{\y}{\y}
1690   \ADD{#2}{\Dy}{\Dy}
1691   \MULTIPLY{\t}{\y}{\y}
1692   \ADD{#1}{\y}{\y}
1693   }
```

## 14.4 Elementary functions

`\ONEfunction` The `\ONEfunction`:  $y(t) = 1$ ,  $y'(t) = 0$

```
1694 \newfunction{\ONEfunction}{%
1695   \COPY{1}{\y}
1696   \COPY{0}{\Dy}}
```

`\ZEROfunction` The `\ZEROfunction`:  $y(t) = 0$ ,  $y'(t) = 0$

```
1697 \newfunction{\ZEROfunction}{%
1698   \COPY{0}{\y}
1699   \COPY{0}{\Dy}}
```

`\IDENTITYfunction` The `\IDENTITYfunction`:  $y(t) = t, y'(t) = 1$

```
1700 \newfunction{\IDENTITYfunction}{%
1701     \COPY{t}{\y}
1702     \COPY{1}{\Dy}}
```

`\RECIPROCALfunction` The `\RECIPROCALfunction`:  $y(t) = 1/t, y'(t) = -1/t^2$

```
1703 \newfunction{\RECIPROCALfunction}{%
1704     \DIVIDE{1}{t}{\y}
1705     \SQUARE{\y}{\Dy}
1706     \MULTIPLY{-1}{\Dy}{\Dy}}
```

`\SQUAREfunction` The `\SQUAREfunction`:  $y(t) = t^2, y'(t) = 2t$

```
1707 \newfunction{\SQUAREfunction}{%
1708     \SQUARE{t}{\y}
1709     \MULTIPLY{2}{t}{\Dy}}
```

`\CUBEfunction` The `\CUBEfunction`:  $y(t) = t^3, y'(t) = 3t^2$

```
1710 \newfunction{\CUBEfunction}{%
1711     \SQUARE{t}{\Dy}
1712     \MULTIPLY{t}{\Dy}{\y}
1713     \MULTIPLY{3}{\Dy}{\Dy}}
```

`\SQRTfunction` The `\SQRTfunction`:  $y(t) = \sqrt{t}, y'(t) = 1/(2\sqrt{t})$

```
1714 \newfunction{\SQRTfunction}{%
1715     \SQRT{t}{\y}
1716     \DIVIDE{0.5}{\y}{\Dy}}
```

`\EXPfunction` The `\EXPfunction`:  $y(t) = \exp t, y'(t) = \exp t$

```
1717 \newfunction{\EXPfunction}{%
1718     \EXP{t}{\y}
1719     \COPY{\y}{\Dy}}
```

`\COSfunction` The `\COSfunction`:  $y(t) = \cos t, y'(t) = -\sin t$

```
1720 \newfunction{\COSfunction}{%
1721     \COS{t}{\y}
1722     \SIN{t}{\Dy}
1723     \MULTIPLY{-1}{\Dy}{\Dy}}
```

`\SINfunction` The `\SINfunction`:  $y(t) = \sin t, y'(t) = \cos t$

```
1724 \newfunction{\SINfunction}{%
1725     \SIN{t}{\y}
1726     \COS{t}{\Dy}}
```

`\TANfunction` The `\TANfunction`:  $y(t) = \tan t, y'(t) = 1/(\cos t)^2$

```
1727 \newfunction{\TANfunction}{%
1728     \TAN{t}{\y}
1729     \COS{t}{\Dy}
1730     \SQUARE{\Dy}{\Dy}
1731     \DIVIDE{1}{\Dy}{\Dy}}
```

`\COTfunction` The `\COTfunction`:  $y(t) = \cot t$ ,  $y'(t) = -1/(\sin t)^2$   
1732 `\newfunction{\COTfunction}{%`  
1733 `\COTAN{t}{y}`  
1734 `\SIN{t}{Dy}`  
1735 `\SQUARE{Dy}{Dy}`  
1736 `\DIVIDE{-1}{Dy}{Dy}`

`\COSHfunction` The `\COSHfunction`:  $y(t) = \cosh t$ ,  $y'(t) = \sinh t$   
1737 `\newfunction{\COSHfunction}{%`  
1738 `\COSH{t}{y}`  
1739 `\SINH{t}{Dy}`

`\SINHfunction` The `\SINHfunction`:  $y(t) = \sinh t$ ,  $y'(t) = \cosh t$   
1740 `\newfunction{\SINHfunction}{%`  
1741 `\SINH{t}{y}`  
1742 `\COSH{t}{Dy}`

`\TANHfunction` The `\TANHfunction`:  $y(t) = \tanh t$ ,  $y'(t) = 1/(\cosh t)^2$   
1743 `\newfunction{\TANHfunction}{%`  
1744 `\TANH{t}{y}`  
1745 `\COSH{t}{Dy}`  
1746 `\SQUARE{Dy}{Dy}`  
1747 `\DIVIDE{1}{Dy}{Dy}`

`\COTHfunction` The `\COTHfunction`:  $y(t) = \coth t$ ,  $y'(t) = -1/(\sinh t)^2$   
1748 `\newfunction{\COTHfunction}{%`  
1749 `\COTANH{t}{y}`  
1750 `\SINH{t}{Dy}`  
1751 `\SQUARE{Dy}{Dy}`  
1752 `\DIVIDE{-1}{Dy}{Dy}`

`\LOGfunction` The `\LOGfunction`:  $y(t) = \log t$ ,  $y'(t) = 1/t$   
1753 `\newfunction{\LOGfunction}{%`  
1754 `\LOG{t}{y}`  
1755 `\DIVIDE{1}{t}{Dy}`

`\HEAVISIDEfunction` The `\HEAVISIDEfunction`:  $y(t) = \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{if } t \geq 0 \end{cases}$ ,  $y'(t) = 0$   
1756 `\newfunction{\HEAVISIDEfunction}{%`  
1757 `\ifdim t\p@<z@ \COPY{0}{y}\else\COPY{1}{y}\fi`  
1758 `\COPY{0}{Dy}`

`\ARCSINfunction` The `\ARCSINfunction`:  $y(t) = \arcsin t$ ,  $y'(t) = 1/\sqrt{1-t^2}$   
1759 `\newfunction{\ARCSINfunction}{%`  
1760 `\ARCSIN{t}{y}`  
1761 `\SQUARE{t}{yy}`  
1762 `\SUBTRACT{1}{yy}{yy}`  
1763 `\SQRT{yy}{Dy}`  
1764 `\DIVIDE{1}{Dy}{Dy}`

`\ARCCOSfunction` The `\ARCCOSfunction`:  $y(t) = \arccos t$ ,  $y'(t) = -1/\sqrt{1-t^2}$

```

1765 \newfunction{\ARCCOSfunction}{%
1766     \ARCCOS{\t}{\y}
1767     \SQUARE{\t}{\yy}
1768     \SUBTRACT{1}{\yy}{\yy}
1769     \SQRT{\yy}{\Dy}
1770     \DIVIDE{-1}{\Dy}{\Dy}}

```

`\ARCTANfunction` The `\ARCTANfunction`:  $y(t) = \arctan t$ ,  $y'(t) = 1/(1+t^2)$

```

1771 \newfunction{\ARCTANfunction}{%
1772     \ARCTAN{\t}{\y}
1773     \SQUARE{\t}{\yy}
1774     \ADD{1}{\yy}{\yy}
1775     \DIVIDE{1}{\yy}{\Dy}}

```

`\ARCCOTfunction` The `\ARCCOTfunction`:  $y(t) = \operatorname{arccot} t$ ,  $y'(t) = -1/(1+t^2)$

```

1776 \newfunction{\ARCCOTfunction}{%
1777     \ARCCOT{\t}{\y}
1778     \SQUARE{\t}{\yy}
1779     \ADD{1}{\yy}{\yy}
1780     \DIVIDE{-1}{\yy}{\Dy}}

```

`\ARSINHfunction` The `\ARSINHfunction`:  $y(t) = \operatorname{arsinh} t$ ,  $y'(t) = 1/\sqrt{1+t^2}$

```

1781 \newfunction{\ARSINHfunction}{%
1782     \ARSINH{\t}{\y}
1783     \SQUARE{\t}{\yy}
1784     \ADD{1}{\yy}{\yy}
1785     \SQRT{\yy}{\Dy}
1786     \DIVIDE{1}{\Dy}{\Dy}}

```

`\ARCOSHfunction` The `\ARSINHfunction`:  $y(t) = \operatorname{arcosh} t$ ,  $y'(t) = 1/\sqrt{t^2-1}$

```

1787 \newfunction{\ARCOSHfunction}{%
1788     \ARCOSH{\t}{\y}
1789     \SQUARE{\t}{\yy}
1790     \SUBTRACT{\yy}{1}{\yy}
1791     \SQRT{\yy}{\Dy}
1792     \DIVIDE{1}{\Dy}{\Dy}}

```

`\ARTANHfunction` The `\ARTANHfunction`:  $y(t) = \operatorname{artanh} t$ ,  $y'(t) = 1/(t^2-1)$

```

1793 \newfunction{\ARTANHfunction}{%
1794     \ARTANH{\t}{\y}
1795     \SQUARE{\t}{\yy}
1796     \SUBTRACT{1}{\yy}{\yy}
1797     \DIVIDE{1}{\yy}{\Dy}}

```

`\ARCOTHfunction` The `\ARCOTHfunction`:  $y(t) = \operatorname{arcoth} t$ ,  $y'(t) = 1/(t^2-1)$

```

1798 \newfunction{\ARCOTHfunction}{%
1799     \ARCOTH{\t}{\y}
1800     \SQUARE{\t}{\yy}
1801     \SUBTRACT{1}{\yy}{\yy}
1802     \DIVIDE{1}{\yy}{\Dy}}

```

## 14.5 Operations with functions

`\CONSTANTfunction` `\CONSTANTfunction` defines #2 as the constant function  $f(t) = \#1$ .

```
1803 \def\CONSTANTfunction#1#2{%
1804     \def#2##1##2##3{%
1805         \xdef##2{#1}%
1806         \xdef##3{0}}}
```

`\SUMfunction` `\SUMfunction` defines #3 as the sum of functions #1 and #2.

```
1807 \def\SUMfunction#1#2#3{%
1808     \def#3##1##2##3{%
1809         \begingroup
1810             #1{##1}{\ccls@SUMf}{\ccls@SUMDf}%
1811             #2{##1}{\ccls@SUMg}{\ccls@SUMDg}%
1812             \ADD{\ccls@SUMf}{\ccls@SUMg}{\ccls@SUMfg}
1813             \ADD{\ccls@SUMDf}{\ccls@SUMDg}{\ccls@SUMDfg}
1814             \xdef##2{\ccls@SUMfg}%
1815             \xdef##3{\ccls@SUMDfg}%
1816         \endgroup}\ignorespaces}
```

`\SUBTRACTfunction` `\SUBTRACTfunction` defines #3 as the difference of functions #1 and #2.

```
1817 \def\SUBTRACTfunction#1#2#3{%
1818     \def#3##1##2##3{%
1819         \begingroup
1820             #1{##1}{\ccls@SUBf}{\ccls@SUBDf}%
1821             #2{##1}{\ccls@SUBg}{\ccls@SUBDg}%
1822             \SUBTRACT{\ccls@SUBf}{\ccls@SUBg}{\ccls@SUBfg}
1823             \SUBTRACT{\ccls@SUBDf}{\ccls@SUBDg}{\ccls@SUBDfg}
1824             \xdef##2{\ccls@SUBfg}%
1825             \xdef##3{\ccls@SUBDfg}%
1826         \endgroup}\ignorespaces}
```

`\PRODUCTfunction` `\PRODUCTfunction` defines #3 as the product of functions #1 and #2.

```
1827 \def\PRODUCTfunction#1#2#3{%
1828     \def#3##1##2##3{%
1829         \begingroup
1830             #1{##1}{\ccls@PROf}{\ccls@PRODf}%
1831             #2{##1}{\ccls@PROg}{\ccls@PRODg}%
1832             \MULTIPLY{\ccls@PROf}{\ccls@PROg}{\ccls@PROfg}
1833             \MULTIPLY{\ccls@PROf}{\ccls@PRODg}{\ccls@PROfDg}
1834             \MULTIPLY{\ccls@PRODf}{\ccls@PROg}{\ccls@PRODfg}
1835             \ADD{\ccls@PROfDg}{\ccls@PRODfg}{\ccls@PRODfg}
1836             \xdef##2{\ccls@PROfg}%
1837             \xdef##3{\ccls@PRODfg}%
1838         \endgroup}\ignorespaces}
```

`\QUOTIENTfunction` `\QUOTIENTfunction` defines #3 as the quotient of functions #1 and #2.

```
1839 \def\QUOTIENTfunction#1#2#3{%
1840     \def#3##1##2##3{%
1841         \begingroup
```

```

1842      #1{##1}{\ccls@QUOf}{\ccls@QUODf}%
1843      #2{##1}{\ccls@QUOg}{\ccls@QUODg}%
1844      \DIVIDE{\ccls@QUOf}{\ccls@QUOg}{\ccls@QUOfg}
1845      \MULTIPLY{\ccls@QUOf}{\ccls@QUODg}{\ccls@QUOfDg}
1846      \MULTIPLY{\ccls@QUODf}{\ccls@QUOg}{\ccls@QUODfg}
1847      \SUBTRACT{\ccls@QUODf}{\ccls@QUOfDg}{\ccls@QUOnum}
1848      \SQUARE{\ccls@QUOg}{\ccls@qsquaretempg}
1849      \DIVIDE{\ccls@QUOnum}{\ccls@qsquaretempg}{\ccls@QUODfg}
1850      \xdef##2{\ccls@QUOfg}%
1851      \xdef##3{\ccls@QUODfg}%
1852      \endgroup}\ignorespaces}

```

`\COMPOSITIONfunction` `\COMPOSITIONfunction` defines #3 as the composition of functions #1 and #2.

```

1853 \def\COMPOSITIONfunction#1#2#3{%  #3=#1(#2)
1854     \def##3##1##2##3{%
1855     \begingroup
1856         #2{##1}{\ccls@COMg}{\ccls@COMDg}%
1857         #1{\ccls@COMg}{\ccls@COMf}{\ccls@COMDf}%
1858         \MULTIPLY{\ccls@COMDg}{\ccls@COMDf}{\ccls@COMDf}
1859         \xdef##2{\ccls@COMf}%
1860         \xdef##3{\ccls@COMDf}%
1861     \endgroup}\ignorespaces}

```

`\SCALEfunction` `\SCALEfunction` defines #3 as the product of number #1 and function #2.

```

1862 \def\SCALEfunction#1#2#3{%
1863     \def##3##1##2##3{%
1864     \begingroup
1865         #2{##1}{\ccls@SCFf}{\ccls@SCFDf}%
1866         \MULTIPLY{#1}{\ccls@SCFf}{\ccls@SCFaf}
1867         \MULTIPLY{#1}{\ccls@SCFDf}{\ccls@SCFDaf}
1868         \xdef##2{\ccls@SCFaf}%
1869         \xdef##3{\ccls@SCFDaf}%
1870     \endgroup}\ignorespaces}

```

`\SCALEVARIABLEfunction` `\SCALEVARIABLEfunction` scales the variable by number #1 and applies function #2.

```

1871 \def\SCALEVARIABLEfunction#1#2#3{%
1872     \def##3##1##2##3{%
1873     \begingroup%
1874         \MULTIPLY{#1}{##1}{\ccls@SCVat}
1875         #2{\ccls@SCVat}{\ccls@SCVf}{\ccls@SCVDf}%
1876         \MULTIPLY{#1}{\ccls@SCVDf}{\ccls@SCVDf}
1877         \xdef##2{\ccls@SCVf}%
1878         \xdef##3{\ccls@SCVDf}%
1879     \endgroup}\ignorespaces}

```

`\POWERfunction` `\POWERfunction` defines #3 as the power of function #1 to exponent #2.

```

1880 \def\POWERfunction#1#2#3{%
1881     \def##3##1##2##3{%
1882     \begingroup
1883         #1{##1}{\ccls@POWf}{\ccls@POWDf}%

```

```

1884         \POWER{\ccls@POWf}{#2}{\ccls@POWfn}
1885         \SUBTRACT{#2}{1}{\ccls@nminusone}
1886         \POWER{\ccls@POWf}{\ccls@nminusone}{\ccls@POWdfn}
1887         \MULTIPLY{#2}{\ccls@POWdfn}{\ccls@POWdfn}
1888         \MULTIPLY{\ccls@POWdfn}{\ccls@POWdf}{\ccls@POWdfn}
1889         \xdef##2{\ccls@POWfn}%
1890         \xdef##3{\ccls@POWdfn}%
1891     \endgroup}\ignorespaces}

```

`\LINEARCOMBINATIONfunction` `\LINEARCOMBINATIONfunction` defines the new function #5 as the linear combination #1#2+#3#4. #1 and #3 are two numbers. #1 and #3 are two functions.

```

1892 \def\LINEARCOMBINATIONfunction#1#2#3#4#5{%
1893     \def#5##1##2##3{%
1894         \begingroup
1895             #2{##1}{\ccls@LINF}{\ccls@LINDf}%
1896             #4{##1}{\ccls@LING}{\ccls@LINDg}%
1897             \MULTIPLY{#1}{\ccls@LINF}{\ccls@LINF}
1898             \MULTIPLY{#3}{\ccls@LING}{\ccls@LING}
1899             \MULTIPLY{#1}{\ccls@LINDf}{\ccls@LINDf}
1900             \MULTIPLY{#3}{\ccls@LINDg}{\ccls@LINDg}
1901             \ADD{\ccls@LINF}{\ccls@LING}{\ccls@LINafbg}
1902             \ADD{\ccls@LINDf}{\ccls@LINDg}{\ccls@LINDafbg}
1903             \xdef##2{\ccls@LINafbg}%
1904             \xdef##3{\ccls@LINDafbg}%
1905         \endgroup}\ignorespaces}

```

`\POLARfunction` `\POLARfunction` defines the polar curve #2. #1 is a previously defined function.

```

1906 \def\POLARfunction#1#2{%
1907     \PRODUCTfunction{#1}{\COSfunction}{\ccls@polarx}
1908     \PRODUCTfunction{#1}{\SINfunction}{\ccls@polary}
1909     \PARAMETRICfunction{\ccls@polarx}{\ccls@polary}{#2}

```

`\PARAMETRICfunction` `\PARAMETRICfunction` defines the parametric curve #3. #1 and #2 are the components functions (two previously defined functions).

```

1910 \def\PARAMETRICfunction#1#2#3{%
1911     \def##3##1##2##3##4##5{%
1912         #1{##1}{##2}{##3}
1913         #2{##1}{##4}{##5}}

```

`\VECTORfunction` `\VECTORfunction`: an alias of `\PARAMETRICfunction`.

```

1914 \let\VECTORfunction\PARAMETRICfunction

```

```

1915 % </calculus>

```



## Change History

v1.0		\ARCOTHfunction . . . . .	77
General: First public version . . . . .	1	New commands: \ARSINH, \ARCOSH,	
v1.0a		\ARTANH, \ARCOTH . . . . .	53
General: calculator.dtx modified to make		New commands: \DOTPRODUCT,	
it autoinstallable. calculus.dtx		\VECTORPRODUCT, \CROSSPRODUCT . . .	56
embedded in calculus.dtx . . . . .	1	New commands: \LENGTHADD,	
v2.0		\LENGTHSUBTRACT . . . . .	34
General: new calculator.dtx and		Trivial error in documentation corrected	68
calculator.ins files . . . . .	1		
New commands: \ARCSINfunction,		v2.1	
\ARCCOSfunction, \ARCTANfunction,		\@BASICLOG: Changed stop criterion on	
\ARCCOTfunction . . . . .	76	iterations to 2sp . . . . .	49
New commands: \ARCSIN, \ARCCOS,		\FRACTIONALPART: Bug fixed . . . . .	36
\ARCTAN, \ARCCOT . . . . .	49	\ROUND: Bug fixed . . . . .	37
New commands: \ARSINHfunction,		\TRUNCATE: Bug fixed . . . . .	36
\ARCOSHfunction, \ARTANHfunction,		General: Some bugs fixed . . . . .	1

## 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			
\# . . . . .	737	\@@@TRANPOSEMATRIX . . . . .	1187, 1196
\@@@DETERMINANT . . . . .	1310, 1323	\@@@TRUNCATE . . . . .	293, 295, 299
\@@@INVERSEMATRIX . . . . .	1339, 1378	\@@@VECTORABSVALUE . . . . .	1044, 1051
\@@@MATRIXABSVALUE . . . . .	1232, 1240	\@@@VECTORADD . . . . .	1020, 1029
\@@@MATRIXADD . . . . .	1201, 1212	\@@@VECTORCOPY . . . . .	949, 956
\@@@MATRIXCOPY . . . . .	1145, 1153	\@@@VECTORGLOBALCOPY . . . . .	960, 967
\@@@MATRIXGLOBALCOPY . . . . .	1157, 1165	\@@@VECTORMATRIXPRODUCT . . . . .	1260, 1272
\@@@MATRIXPRODUCT . . . . .	1293, 1304	\@@@VECTORPRODUCT . . . . .	1004, 1014
\@@@MATRIXSUB . . . . .	1217, 1228	\@@@VECTORSUB . . . . .	1033, 1040
\@@@MATRIXVECTORPRODUCT . . . . .	1245, 1255	\@@DEGREESCOS . . . . .	536, <u>601</u>
\@@@OUTPUTMATRIX . . . . .	1170, 1183	\@@DEGREESCOT . . . . .	538, <u>605</u>
\@@@OUTPUTVECTOR . . . . .	972, 979	\@@DEGREESSIN . . . . .	535, <u>599</u>
\@@@ROUND . . . . .	315, 317, 322	\@@DEGREESTAN . . . . .	537, <u>603</u>
\@@@SCALARMATRIXPRODUCT . . . . .	1277, 1288	\@@DETERMINANT . . . . .	1305, 1322
\@@@SCALARPRODUCT . . . . .	985, 997	\@@EXP . . . . .	609, <u>610</u>
\@@@SCALARVECTORPRODUCT . . . . .	1056, 1065	\@@FRACTIONALPART . . . . .	274, 276
\@@@SOLVELINEARSYSTEM . . . . .	1436, 1455	\@@INTEGERPART . . . . .	259, 261
		\@@INVERSEMATRIX . . . . .	1324, 1376
		\@@LOG . . . . .	713, <u>732</u>
		\@@MATRIXABSVALUE . . . . .	1229, 1239
		\@@MATRIXADD . . . . .	1197, 1211
		\@@MATRIXCOPY . . . . .	946, 955
		\@@MATRIXGLOBALCOPY . . . . .	1154, 1164
		\@@MATRIXPRODUCT . . . . .	1289, 1303
		\@@MATRIXSUB . . . . .	1213, 1227
		\@@MATRIXVECTORPRODUCT . . . . .	1241, 1254
		\@@OUTPUTMATRIX . . . . .	1166, 1182
		\@@OUTPUTVECTOR . . . . .	968, 978
		\@@ROUND . . . . .	307–309
		\@@SCALARMATRIXPRODUCT . . . . .	1273, 1287
		\@@SCALARPRODUCT . . . . .	980, 996
		\@@SCALARVECTORPRODUCT . . . . .	1052, 1064
		\@@SOLVELINEARSYSTEM . . . . .	1391, 1453
		\@@TDMATRIXSOL . . . . .	<u>1133</u> , 1354
		\@@TRANPOSEMATRIX . . . . .	1184, 1195
		\@@TRUNCATE . . . . .	285–287
		\@@VECTORABSVALUE . . . . .	1041, 1050
		\@@VECTORADD . . . . .	1016, 1028
		\@@VECTORCOPY . . . . .	946, 955



\ccls@polary . . . . .	1908, 1909	\cctr@@detc . . . . .	1397, 1405	\cctr@outd . . . . .	
\ccls@POWdf . . . . .	1883, 1888	\cctr@absval . . . . .	622,	. 1167, 1168, 1173, 1177	
\ccls@POWdfn 1886–1888,	1890	623, 667, 668, 680,		\cctr@oute . . . . .	1173, 1177
\ccls@POWf . 1883, 1884,	1886	681, 693, 694, 704, 705		\cctr@outf . . . . .	1173, 1177
\ccls@POWfn . . . . .	1884, 1889	\cctr@ae . . 721, 722, 726, 727		\cctr@outg . . . . .	1174, 1178
\ccls@PROdf . . . . .	1830, 1834	\cctr@det . . . 1311–1315,		\cctr@outh . . . . .	1174, 1178
\ccls@PROdfg 1834, 1835,	1837	1335, 1336, 1347,		\cctr@outi . . . . .	1174, 1178
\ccls@PRODg . . . . .	1831, 1833	1349, 1351, 1372,		\cctr@Q . . . . .	175, 185, 187
\ccls@PROf . 1830, 1832,	1833	1373, 1437, 1438, 1446		\cctr@sign . . . 169–172, 187	
\ccls@PROfdg . . . . .	1833, 1835	\cctr@deta . . . . .		\cctr@size . . . . .	953,
\ccls@PROfg . . . . .	1832, 1836	. 1392, 1395, 1399–1401		954, 964, 965, 976,	
\ccls@PROg . 1831, 1832,	1834	\cctr@detb . 1393, 1396, 1399		977, 994, 995, 1011,	
\ccls@qpoly . . . . .	1652,	\cctr@detc . 1394, 1397, 1400		1012, 1026, 1027,	
1655, 1658, 1661, <u>1662</u>		\cctr@epsilon . . . . . <u>6</u> ,		1037, 1038, 1048,	
\ccls@qsquaretempg 1848,	1849	396, 612, 716, 741,		1049, 1062, 1063,	
\ccls@QUODf . . . . .	1842, 1846	742, 795, 1325,		1150, 1151, 1162,	
\ccls@QUODfg . . . . .		1340, 1398, 1402,		1163, 1180, 1181,	
. 1846, 1847, 1849, 1851		1405, 1410, 1414,		1193, 1194, 1209,	
\ccls@QUODg . . . . .	1843, 1845	1418, 1422, 1427, 1439		1210, 1225, 1226,	
\ccls@QUOf . 1842, 1844,	1845	\cctr@expminux . . . . .		1237, 1238, 1252,	
\ccls@QUOfDg . . . . .	1845, 1847	. . . . 674, 675, 687, 688		1253, 1269, 1270,	
\ccls@QUOfg . . . . .	1844, 1850	\cctr@expt 631, 632, 639, 640		1285, 1286, 1301,	
\ccls@QUOg . . . . .		\cctr@expx 672, 675, 685, 688		1302, 1320, 1321,	
. 1843, 1844, 1846, 1848		\cctr@expy 632, 633, 640, 641		1374, 1375, 1451, 1452	
\ccls@QUOnum . . . . .	1847, 1849	\cctr@lengtha . . . . .		\cctr@sol . . . . .	
\ccls@SCFaf . . . . .	1866, 1868	. . . 4, 157–160, 162,		. 1138, 1311, 1313, 1315	
\ccls@SCFDaf . . . . .	1867, 1869	163, 194, 200, 201,		\cctr@solAA . . . . .	
\ccls@SCFDf . . . . .	1865, 1867	203, 205, 206, 211, 213		. . 1093, 1103, 1113,	
\ccls@SCFf . . . . .	1865, 1866	\cctr@lengthb . . . . .		1134, 1202, 1218,	
\ccls@SCVat . . . . .	1874, 1875	5, 161, 162, 195, 196,		1278, 1294, 1341,	
\ccls@SCVdf . 1875, 1876,	1878	200, 204, 205, 212,		1347, 1348, 1360, 1443	
\ccls@SCVf . . . . .	1875, 1877	214, 395, 396, 403,		\cctr@solAB . . 1094, 1104,	
\ccls@sint . 1594, 1595,	1598	405, 740, 742, 749,		1114, 1202, 1218,	
\ccls@SUBdf . . . . .	1820, 1823	751, 793, 795, 806, 808		1278, 1294, 1341,	
\ccls@SUBdfg . . . . .	1823, 1825	\cctr@log . . . . . 616–618		1347, 1348, 1363, 1443	
\ccls@SUBDg . . . . .	1821, 1823	\cctr@loga . . . . . 733, 735		\cctr@solAC . . 1095, 1105,	
\ccls@SUBf . . . . .	1820, 1822	\cctr@logmaxnum . <u>8</u> , 114,		1115, 1202, 1218,	
\ccls@SUBfg . . . . .	1822, 1824	623, 668, 681, 694, 705		1278, 1294, 1341,	
\ccls@SUBg . . . . .	1821, 1822	\cctr@logx . . . . . 734, 735		1347, 1348, 1366, 1443	
\ccls@SUMdf . . . . .	1810, 1813	\cctr@minust 673, 674, 686, 687		\cctr@solBA . . 1096, 1106,	
\ccls@SUMdfg . . . . .	1813, 1815	\cctr@ndec . . . 176, 177, 186		1116, 1203, 1219,	
\ccls@SUMDg . . . . .	1811, 1813	\cctr@outa 144, 145, 147,		1279, 1295, 1342,	
\ccls@SUMf . . . . .	1810, 1812	969, 970, 973, 974,		1349, 1350, 1361, 1444	
\ccls@SUMfg . . . . .	1812, 1814	1167, 1168, 1172, 1176		\cctr@solBB . . 1097, 1107,	
\ccls@SUMg . . . . .	1811, 1812	\cctr@outb . . . 146, 147,		1117, 1203, 1219,	
\ccls@temp . . . . .	1685, 1686	969, 970, 973, 974,		1279, 1295, 1342,	
\cctr@@det . . . . .	1325,	1167, 1168, 1172, 1176		1349, 1350, 1364, 1444	
1340, 1373, 1438, 1439		\cctr@outc . . . 973, 974,		\cctr@solBC . . 1098, 1108,	
\cctr@@deta . . . . .	1395, 1398	1167, 1168, 1172, 1176		1118, 1203, 1219,	
\cctr@@detb . . . . .	1396, 1402			1279, 1295, 1342,	

1349, 1350, 1367, 1444	584, 835, 836, 840,	\cctr@Warnbigartanh . . . .
\cctr@solCA .. 1099, 1109,	841, 885–887, 897–	..... 60, 905, 909
1119, 1204, 1220,	899, 912, 914, 929,	\cctr@Warncrossprod 131, 1002
1280, 1296, 1343,	931, 1078, 1084, 1089,	\cctr@Warndivzero .. 21, 198
1351, 1352, 1362, 1445	1262–1265, 1408, 1414	\cctr@WarnIncLinSys 83, 1379
\cctr@solCB .. 1100, 1110,	\cctr@tempC 915, 916, 932, 933	\cctr@WarnIncTDLinSys ..
1120, 1204, 1220,	\cctr@tempc ..... 231,	..... 86, 1386
1280, 1296, 1343,	232, 351, 353, 364,	\cctr@WarnIndLinSys 90, 1382
1351, 1352, 1365, 1445	366, 451, 452, 564,	\cctr@Warninfcotan .....
\cctr@solCC .. 1101, 1111,	565, 913, 915, 930,	..... 104, 506, 594
1121, 1204, 1220,	932, 1079, 1088–1090,	\cctr@Warninfexp .. 110,
1280, 1296, 1343,	1262–1265, 1409, 1418	624, 669, 682, 695, 706
1351, 1352, 1368, 1445	\cctr@tempD .....	\cctr@Warninfexpb . 118, 613
\cctr@tanhden .....	. 167, 174, 179, 181, 184	\cctr@Warninflog .. 125, 717
.... 699, 700, 709, 711	\cctr@tempd 168, 174, 182,	\cctr@WarninfTan .....
\cctr@tanhnum .....	184, 242, 251, 1409, 1422	.. 98, 456, 460, 569, 573
.... 698, 700, 710, 711	\cctr@tempdif ..... 748, 749	\cctr@Warnnoangle .....
\cctr@temp ..... 255,	\cctr@tempexp ..... 221, 222	..... 137, 1082, 1086
258, 264–266, 1068, 1069	\cctr@tempm ... 340, 344, 345	\cctr@Warnnogcd .... 26, 356
\cctr@tempA ..... 279, 280	\cctr@tempn ..... 334,	\cctr@Warnnointexp .. 36, 223
\cctr@tempa 213, 215, 234,	337, 338, 340, 343–345	\cctr@Warnnuposrad .. 30, 392
236, 238, 291–293,	\cctr@tempo ..... 329,	\cctr@Warnround .... 15, 311
313–315, 349, 351,	331, 332, 334, 336–	\cctr@Warnsingmatrix 73, 1330
352, 354, 362–366,	338, 340, 342, 343, 345	\cctr@WarnsingTDMatrix .
433, 434, 436, 438,	\cctr@tempoldw .... 743, 748	..... 78, 1344
439, 441, 444, 475,	\cctr@tempoldy .... 797, 807	\cctr@Warnsmallarcosh 54, 893
476, 478, 482, 483,	\cctr@temppp ..... 324–	\cctr@Warnsmallarcoth 66, 925
485, 486, 488, 489,	327, 329–332, 334–	\cctr@Warntruncate .. 9, 289
491, 492, 494, 495,	336, 338, 340–342, 345	\cctr@WarnZeroLinSys 94, 1431
646, 647, 649, 652,	\cctr@tempq .... 174, 175,	\COMPOSITIONfunction . <u>1853</u>
655, 658, 851, 852,	184, 185, 325–327,	\CONSTANTfunction ... <u>1803</u>
854, 857, 860, 863,	329, 330, 332, 334,	\COPY ..... <u>142</u> , 144, 147,
866, 869, 872, 883–	335, 338, 340, 341, 345	150, 153, 156, 169–
885, 895–897, 911–	\cctr@tempR .....	172, 175, 176, 181,
913, 928–930, 939,	. 174, 178, 179, 181, 184	185, 201, 234, 235,
940, 982, 983, 987–	\cctr@tempw .... 397–399,	259, 268, 274, 289,
990, 1072–1074, 1077,	402, 403, 739, 743,	301–305, 311, 324–
1080, 1088, 1262–	744, 746–748, 753,	327, 329–332, 334–
1265, 1307, 1308,	796, 798, 800, 802, 804	338, 340–345, 353,
1408, 1410, 1426, 1427	\cctr@tempX 774–776, 782–784	359, 363, 365, 366,
\cctr@tempB 914, 916, 931, 933	\cctr@tempXw ..... 744–746	370, 378, 388, 394,
\cctr@tempb .....	\cctr@tempXX 776, 777, 784, 785	397, 401, 411, 413,
. 214, 215, 230, 232,	\cctr@tempY 790, 791, 794,	415, 434, 436, 476,
350–353, 355, 359,	797, 800, 801, 804–807	479, 502, 514, 523,
364, 365, 418, 419,	\cctr@tempZ .....	525, 534, 541, 543,
425, 426, 428, 429,	401, 798, 799, 801–803	545, 590, 743, 753,
464, 465, 470, 471,	\cctr@Warnbigarccos ....	757, 760, 763, 789,
548, 549, 552, 553,	..... 48, 823, 827	794, 796, 797, 803,
556, 557, 559, 560,	\cctr@Warnbigarcsin ....	813, 816, 819, 849,
577, 580, 581, 583,	..... 42, 767, 771	911, 928, 941, 944,

945, 947, 950, 1093–1101, 1103–1111, 1138, 1140, 1141, 1143, 1185, 1331, 1332, 1384, 1432, 1649, 1695, 1696, 1698, 1699, 1701, 1702, 1719, 1757, 1758	874, 917, 934, 1073, 1088, 1089, 1335, 1357, 1383, 1399, 1400, 1459, 1461–1464, 1466, 1704, 1716, 1731, 1736, 1747, 1752, 1755, 1764, 1770, 1775, 1780, 1786, 1792, 1797, 1802, 1844, 1849	<code>\forcecpoly</code> . . . . . <a href="#">1659</a>
<code>\COS</code> . . . . . <a href="#">449</a> , 500, 800, 1592, 1721, 1726, 1729	<code>\DOTPRODUCT</code> . . . . . <a href="#">998</a>	<code>\forcevectorfunction</code> . . . . . <a href="#">1622</a>
<code>\COSfunction</code> . . . . . <a href="#">1720</a> , 1907	<code>\Dr</code> . . . . . 1596, 1598	<code>\FRACTIONALPART</code> . . . . . <a href="#">274</a>
<code>\COSH</code> . . . . . <a href="#">665</a> , 699, 710, 1738, 1742, 1745	<code>\Dx</code> . . . . . 1596, 1597, 1601, 1630	<code>\FRACTIONSIMPLIFY</code> . . . . . <a href="#">376</a>
<code>\COSHfunction</code> . . . . . <a href="#">1737</a>	<code>\Dy</code> . . . . . 1566, 1598, 1599, 1603, 1632, 1649, 1664, 1665, 1683, 1686, 1688, 1690, 1696, 1699, 1702, 1705, 1706, 1709, 1711–1713, 1716, 1719, 1722, 1723, 1726, 1729–1731, 1734–1736, 1739, 1742, 1745–1747, 1750–1752, 1755, 1758, 1763, 1764, 1769, 1770, 1775, 1780, 1785, 1786, 1791, 1792, 1797, 1802	<b>G</b>
<code>\COT</code> . . . . . <a href="#">498</a>	<code>\EXP</code> . . . . . <a href="#">609</a> , 672, 674, 685, 687, 744, 1718	<code>\GCD</code> . . . . . <a href="#">347</a> , 369, 380
<code>\COTAN</code> . . . . . <a href="#">1733</a>	<code>\EXPfunction</code> . . . . . <a href="#">1717</a>	<code>\GLOBALCOPY</code> . . . . . . . . . . <a href="#">143</a> , 144–146, 958, 961, 1113–1121, 1155
<code>\COTANH</code> . . . . . <a href="#">1749</a>	<b>F</b>	<code>\HEAVISIDEfunction</code> . . . . . <a href="#">1756</a>
<code>\COTfunction</code> . . . . . <a href="#">1732</a>	<code>\FLOOR</code> . . . . . <a href="#">273</a>	<b>I</b>
<code>\COTH</code> . . . . . <a href="#">702</a>	<code>\forcecpoly</code> . . . . . <a href="#">1678</a>	<code>\IDENTITYfunction</code> . . . . . <a href="#">1700</a>
<code>\COTHfunction</code> . . . . . <a href="#">1748</a>	<code>\forcefunction</code> . . . . . . . . . . <a href="#">1558</a> , 1644, 1660, 1679	<code>\ifcase</code> . . . . . 300, 323
<code>\CROSSPRODUCT</code> . . . . . <a href="#">1015</a>	<code>\forcepolarfunction</code> . . . . . <a href="#">1585</a>	<code>\INTEGERDIVISION</code> <a href="#">240</a> , 255, 258
<code>\CUBE</code> . . . . . <a href="#">218</a>		<code>\INTEGERPART</code> . . . . . 221, 259, 273, 279, 291, 313
<code>\CUBEfunction</code> . . . . . <a href="#">1710</a>		<code>\INTEGERQUOTIENT</code> . . . . . <a href="#">256</a>
<b>D</b>		<code>\INVERSEMATRIX</code> . . . . . <a href="#">1324</a>
<code>\DEGREESCOS</code> . . . . . <a href="#">536</a> , 588, 602	<b>E</b>	<b>L</b>
<code>\DEGREESCOT</code> . . . . . <a href="#">538</a> , 606	<code>\ensurecpoly</code> . . . . . <a href="#">1675</a>	<code>\LCM</code> . . . . . <a href="#">368</a>
<code>\DEGREESSIN</code> . . . . . . . . . . <a href="#">535</a> , 549, 557, 560, 565	<code>\ensurefunction</code> . . . . . 1498, 1553, 1641, 1657, 1676	<code>\LENGTHADD</code> . . . . . <a href="#">203</a> , 208
<code>\DEGREESTAN</code> . . . . . . . . . . <a href="#">537</a> , 577, 584, 592, 604	<code>\ensurepolarfunction</code> . . . . . 1519, <a href="#">1580</a>	<code>\LENGTHDIVIDE</code> . . . . . <a href="#">209</a>
<code>\DEGtoRAD</code> . . . . . <a href="#">511</a> , 552, 580	<code>\ensurepolarfunction</code> . . . . . 1519, <a href="#">1580</a>	<code>\LENGTHSUBTRACT</code> . . . . . <a href="#">207</a>
<code>\DETERMINANT</code> . . . . . 1005–1007, <a href="#">1305</a> , 1360–1368, 1372, 1392–1394, 1437	<code>\ensurepolarfunction</code> . . . . . 1519, <a href="#">1580</a>	<code>\LINEARCOMBINATIONfunction</code> . . . . . <a href="#">1892</a>
<code>\DIVIDE</code> . . . . . <a href="#">165</a> , 215, 230, 372, 381, 382, 398, 400, 439, 442, 445, 481, 482, 484, 485, 487, 488, 490, 491, 493, 494, 496, 509, 511, 597, 607, 631, 639, 647, 649, 650, 652, 653, 655, 656, 658, 659, 662, 676, 689, 700, 711, 721, 735, 745, 775, 783, 801, 835, 840, 854, 857, 860, 863, 866, 869, 872,	<code>\ensurepolarfunction</code> . . . . . 1519, <a href="#">1580</a>	<code>\LOG</code> 616, <a href="#">713</a> , 727, 887, 899, 914, 915, 931, 932, 1754
	<code>\ensurepolarfunction</code> . . . . . 1519, <a href="#">1580</a>	<code>\LOGfunction</code> . . . . . <a href="#">1753</a>
	<code>\EXP</code> . . . . . <a href="#">609</a> , 672, 674, 685, 687, 744, 1718	<b>M</b>
	<code>\EXPfunction</code> . . . . . <a href="#">1717</a>	<code>\MATRIXABSVALUE</code> . . . . . <a href="#">1229</a> , 1408
	<b>F</b>	<code>\MATRIXADD</code> . . . . . <a href="#">1197</a>
	<code>\FLOOR</code> . . . . . <a href="#">273</a>	<code>\MATRIXCOPY</code> . . . . . <a href="#">1142</a> , 1168, 1175
	<code>\forcecpoly</code> . . . . . <a href="#">1678</a>	<code>\MATRIXGLOBALCOPY</code> . . . . . . . . . . <a href="#">1154</a> , 1167, 1171
	<code>\forcefunction</code> . . . . . . . . . . <a href="#">1558</a> , 1644, 1660, 1679	<code>\MATRIXPRODUCT</code> . . . . . <a href="#">1289</a>
	<code>\forcepolarfunction</code> . . . . . <a href="#">1585</a>	<code>\MATRIXSIZE</code> . . . . . <a href="#">1139</a> , 1150, 1162, 1180, 1193, 1209, 1225, 1237, 1252, 1285, 1301, 1320, 1374, 1451
		<code>\MATRIXSUB</code> . . . . . <a href="#">1213</a>
		<code>\MATRIXVECTORPRODUCT</code> . . . . . . . . . . <a href="#">1241</a> , 1290, 1291, 1443
		<code>\MAX</code> . . . . . <a href="#">148</a> , 351
		<code>\MIN</code> . . . . . <a href="#">151</a> , 352

<code>\MODULO</code> . . . . .	<a href="#">253</a> , <a href="#">364</a>	<code>\numberFIFTHPI</code> . . . . .	<a href="#">1463</a>	<b>R</b>	
<code>\MULTIPLY</code> . . . . .	<a href="#">156</a> , <a href="#">157</a> , <a href="#">179</a> , <a href="#">182</a> , <a href="#">187</a> , <a href="#">191</a> , <a href="#">217</a> , <a href="#">218</a> , <a href="#">231</a> , <a href="#">237</a> , <a href="#">264</a> , <a href="#">373</a> , <a href="#">383</a> , <a href="#">441</a> , <a href="#">444</a> , <a href="#">447</a> , <a href="#">512</a> , <a href="#">608</a> , <a href="#">617</a> , <a href="#">663</a> , <a href="#">673</a> , <a href="#">686</a> , <a href="#">726</a> , <a href="#">778</a> , <a href="#">786</a> , <a href="#">852</a> , <a href="#">855</a> , <a href="#">858</a> , <a href="#">861</a> , <a href="#">864</a> , <a href="#">867</a> , <a href="#">870</a> , <a href="#">939</a> , <a href="#">981</a> , <a href="#">982</a> , <a href="#">986</a> , <a href="#">987</a> , <a href="#">989</a> , <a href="#">1053</a> , <a href="#">1054</a> , <a href="#">1057–1059</a> , <a href="#">1306</a> , <a href="#">1307</a> , <a href="#">1311</a> , <a href="#">1312</a> , <a href="#">1314</a> , <a href="#">1333</a> , <a href="#">1334</a> , <a href="#">1458</a> , <a href="#">1460</a> , <a href="#">1593</a> , <a href="#">1595</a> , <a href="#">1596</a> , <a href="#">1598</a> , <a href="#">1647</a> , <a href="#">1663</a> , <a href="#">1664</a> , <a href="#">1667</a> , <a href="#">1682</a> , <a href="#">1683</a> , <a href="#">1685</a> , <a href="#">1687</a> , <a href="#">1688</a> , <a href="#">1691</a> , <a href="#">1706</a> , <a href="#">1709</a> , <a href="#">1712</a> , <a href="#">1713</a> , <a href="#">1723</a> , <a href="#">1832–1834</a> , <a href="#">1845</a> , <a href="#">1846</a> , <a href="#">1858</a> , <a href="#">1866</a> , <a href="#">1867</a> , <a href="#">1874</a> , <a href="#">1876</a> , <a href="#">1887</a> , <a href="#">1888</a> , <a href="#">1897–1900</a>	<code>\numberGOLD</code> . . . . .	<a href="#">1470</a>	<code>\r</code> . . . . .	<a href="#">1593</a> , <a href="#">1595</a>
		<code>\numberHALFPI</code> . . . . .	<a href="#">411</a> , <a href="#">413</a> , <a href="#">417</a> , <a href="#">421</a> , <a href="#">451</a> , <a href="#">455</a> , <a href="#">459</a> , <a href="#">463</a> , <a href="#">467</a> , <a href="#">760</a> , <a href="#">763</a> , <a href="#">779</a> , <a href="#">787</a> , <a href="#">813</a> , <a href="#">830</a> , <a href="#">837</a> , <a href="#">842</a> , <a href="#">879</a> , <a href="#">1459</a>	<code>\RADtoDEG</code> . . . . .	<a href="#">512</a>
		<code>\numberINVE</code> . . . . .	<a href="#">1466</a> , <a href="#">1468</a>	<code>\RECIPROCALfunction</code> . . . . .	<a href="#">1703</a>
		<code>\numberINVETWO</code> . . . . .	<a href="#">1468</a>	<code>\REDUCEDEGREESANGLE</code> . . . . .	<a href="#">524</a>
		<code>\numberINVGOLD</code> . . . . .	<a href="#">1471</a>	<code>\REDUCERADIANSANGLE</code> . . . . .	<a href="#">513</a>
		<code>\numberLOGTEN</code> . . . . .	<a href="#">1469</a>	<code>\renewcpoly</code> . . . . .	<a href="#">1672</a>
		<code>\numberPI</code> . . . . .	<a href="#">425</a> , <a href="#">464</a> , <a href="#">470</a> , <a href="#">515</a> , <a href="#">519</a> , <a href="#">523</a> , <a href="#">819</a> , <a href="#">1457</a> , <a href="#">1458–1464</a>	<code>\renewfunction</code> . . . . .	<a href="#">1488</a> , <a href="#">1547</a> , <a href="#">1638</a> , <a href="#">1654</a> , <a href="#">1673</a>
		<code>\numberQUARTERPI</code> . . . . .	<a href="#">1462</a>	<code>\renewlpoly</code> . . . . .	<a href="#">1637</a>
		<code>\numberSIXTHPI</code> . . . . .	<a href="#">1464</a>	<code>\renewpolarfunction</code> . . . . .	<a href="#">1505</a> , <a href="#">1574</a>
		<code>\numbersQRTFIVE</code> . . . . .	<a href="#">1474</a>	<code>\renewqpoly</code> . . . . .	<a href="#">1653</a>
		<code>\numbersQRTTHREE</code> . . . . .	<a href="#">1473</a>	<code>\renewvectorfunction</code> . . . . .	<a href="#">1526</a> , <a href="#">1611</a>
		<code>\numbersQRTTWO</code> . . . . .	<a href="#">1472</a>	<code>\ROUND</code> . . . . .	<a href="#">307</a>
		<code>\numberTHIRDPI</code> . . . . .	<a href="#">1461</a>	<b>S</b>	
		<code>\numberTHREEHALFPI</code> . . . . .	<a href="#">415</a> , <a href="#">424</a> , <a href="#">1460</a>	<code>\SCALARMATRIXPRODUCT</code> . . . . .	<a href="#">1273</a> , <a href="#">1336</a>
		<code>\numberTWOPI</code> . . . . .	<a href="#">418</a> , <a href="#">428</a> , <a href="#">516</a> , <a href="#">520</a> , <a href="#">1458</a>	<code>\SCALARPRODUCT</code> . . . . .	<a href="#">980</a> , <a href="#">998</a> , <a href="#">1068</a> , <a href="#">1079</a> , <a href="#">1242</a> , <a href="#">1243</a> , <a href="#">1246–1248</a> , <a href="#">1257</a> , <a href="#">1258</a>
		<b>O</b>		<code>\SCALARVECTORPRODUCT</code> . . . . .	<a href="#">1052</a> , <a href="#">1074</a> , <a href="#">1261</a> , <a href="#">1262</a> , <a href="#">1264</a> , <a href="#">1274</a> , <a href="#">1275</a> , <a href="#">1278–1280</a>
	<b>N</b>	<code>\ONEfunction</code> . . . . .	<a href="#">1694</a>	<code>\SCALEfunction</code> . . . . .	<a href="#">1862</a>
	<code>\NeedsTeXFormat</code> . . . . .	<code>\or</code> <a href="#">302–305</a> , <a href="#">325</a> , <a href="#">329</a> , <a href="#">334</a> , <a href="#">340</a>		<code>\SCALEVARIABLEfunction</code> . . . . .	<a href="#">1871</a>
	<code>\newcpoly</code> . . . . .	<b>P</b>		<code>\setlength</code> . . . . .	<a href="#">206</a>
	<code>\newfunction</code> . . . . .	<code>\PackageError</code> <a href="#">1483</a> , <a href="#">1490</a> , <a href="#">1500</a> , <a href="#">1508</a> , <a href="#">1521</a> , <a href="#">1529</a>		<code>\SIN</code> . . . . .	<a href="#">409</a> , <a href="#">452</a> , <a href="#">798</a> , <a href="#">1594</a> , <a href="#">1722</a> , <a href="#">1725</a> , <a href="#">1734</a>
		<code>\PackageInfo</code> <a href="#">1496</a> , <a href="#">1517</a> , <a href="#">1538</a>		<code>\SINfunction</code> . . . . .	<a href="#">1724</a> , <a href="#">1908</a>
		<code>\PackageWarning</code> . . . . .	<a href="#">10</a> , <a href="#">16</a> , <a href="#">22</a> , <a href="#">27</a> , <a href="#">31</a> , <a href="#">37</a> , <a href="#">43</a> , <a href="#">49</a> , <a href="#">55</a> , <a href="#">61</a> , <a href="#">67</a> , <a href="#">74</a> , <a href="#">79</a> , <a href="#">83</a> , <a href="#">86</a> , <a href="#">90</a> , <a href="#">94</a> , <a href="#">99</a> , <a href="#">105</a> , <a href="#">111</a> , <a href="#">119</a> , <a href="#">126</a> , <a href="#">132</a> , <a href="#">138</a>	<code>\SINH</code> . . . . .	<a href="#">678</a> , <a href="#">698</a> , <a href="#">709</a> , <a href="#">1739</a> , <a href="#">1741</a> , <a href="#">1750</a>
		<code>\PARAMETRICfunction</code> . . . . .	<a href="#">1909</a> , <a href="#">1910</a> , <a href="#">1914</a>	<code>\SINHfunction</code> . . . . .	<a href="#">1740</a>
		<code>\POLARfunction</code> . . . . .	<a href="#">1906</a>	<code>\SOLVELINEARSYSTEM</code> . . . . .	<a href="#">1379</a>
		<code>\POWER</code> . . . . .	<a href="#">219</a> , <a href="#">1884</a> , <a href="#">1886</a>	<code>\SQRT</code> . . . . .	<a href="#">408</a> , <a href="#">776</a> , <a href="#">784</a> , <a href="#">885</a> , <a href="#">897</a> , <a href="#">1715</a> , <a href="#">1763</a> , <a href="#">1769</a> , <a href="#">1785</a> , <a href="#">1791</a>
		<code>\POWERfunction</code> . . . . .	<a href="#">1880</a>	<code>\SQRTfunction</code> . . . . .	<a href="#">1714</a>
		<code>\PRODUCTfunction</code> . . . . .	<a href="#">1827</a> , <a href="#">1907</a> , <a href="#">1908</a>	<code>\SQUARE</code> . . . . .	<a href="#">217</a> , <a href="#">438</a> , <a href="#">633</a> , <a href="#">641</a> , <a href="#">646</a> , <a href="#">851</a> , <a href="#">883</a> , <a href="#">895</a> , <a href="#">1467</a> , <a href="#">1468</a> , <a href="#">1705</a> , <a href="#">1708</a> , <a href="#">1711</a> , <a href="#">1730</a> , <a href="#">1735</a> , <a href="#">1746</a> , <a href="#">1751</a> , <a href="#">1761</a> , <a href="#">1767</a> , <a href="#">1773</a> , <a href="#">1778</a> , <a href="#">1783</a> , <a href="#">1789</a> , <a href="#">1795</a> , <a href="#">1800</a> , <a href="#">1848</a>
		<code>\ProvidesPackage</code> . . . . .	<a href="#">3</a> , <a href="#">1480</a>		
		<b>Q</b>			
	<code>\numberE</code> . . . . .	<code>\QUOTIENTfunction</code> . . . . .	<a href="#">1839</a>		
	<a href="#">721</a> , <a href="#">726</a> , <a href="#">1465</a> , <a href="#">1466</a> , <a href="#">1467</a>				
	<code>\numberETWO</code> . . . . .				
	<a href="#">720</a> , <a href="#">1467</a>				

<code>\SQUAREfunction</code> . . . . .	<a href="#">1707</a>	1754, 1755, 1757,	<code>\VECTORSIZE</code> <a href="#">943</a> , 953, 964,
<code>\SQUAREROOT</code> ..	<a href="#">385</a> , 408, 1069	1760, 1761, 1766,	976, 994, 1011, 1026,
<code>\SUBTRACT</code> . . . . .	<a href="#">164</a> ,	1767, 1772, 1773,	1037, 1048, 1062, 1269
192, 249, 267, 280,		1777, 1778, 1782,	<code>\VECTORSUB</code> . . . . .
402, 425, 428, 440,		1783, 1788, 1789,	<a href="#">1030</a> ,
443, 446, 470, 483,		1794, 1795, 1799, 1800	1214, 1215, 1218–1220
486, 489, 492, 495,		<code>\TAN</code> . . . . .	<b>X</b>
520, 531, 556, 559,		<a href="#">453</a> , 504, 1728	<code>\x</code> . . .
583, 661, 688, 728,		<code>\TANfunction</code> . . . . .	1593, 1599, 1600, 1629
739, 747, 748, 774,		<a href="#">1727</a>	<b>Y</b>
779, 787, 799, 802,		<code>\TANH</code> . . . . .	<code>\y</code> . . . .
804, 830, 837, 842,		<a href="#">691</a> , 1744	1565, 1595, 1597,
879, 896, 913, 916,		<code>\TANHfunction</code> . . . . .	1602, 1631, 1647,
930, 933, 1308,		<a href="#">1743</a>	1648, 1663, 1664,
1597, 1762, 1768,		<code>\textit</code> . . . . .	1666–1668, 1682–
1790, 1796, 1801,		<a href="#">737</a>	1684, 1687, 1689,
1822, 1823, 1847, 1885		<code>\TRANPOSEMATRIX</code> . . . .	1691, 1692, 1695,
<code>\SUBTRACTfunction</code> . . .	<a href="#">1817</a>	<a href="#">1184</a>	1698, 1701, 1704,
<code>\SUMfunction</code> . . . . .	<a href="#">1807</a>	<code>\TRUNCATE</code> . . . . .	1705, 1708, 1712,
		<a href="#">285</a>	1715, 1716, 1718,
		<code>\TWOVECTORSANGLE</code> . . . .	1719, 1721, 1725,
		<a href="#">1075</a>	1728, 1733, 1738,
		<b>U</b>	1741, 1744, 1749,
		<code>\UNITVECTOR</code> . . . . .	1754, 1757, 1760,
		<a href="#">1070</a>	1766, 1772, 1777,
		<b>V</b>	1782, 1788, 1794, 1799
		<code>\VECTORABSVALUE</code> . . . . .	<code>\yy</code> . . .
		. . . . .	1761–1763, 1767–
		<a href="#">1041</a> , 1230, 1233	1769, 1773–1775,
<code>\t</code> . . . .	1563, 1590, 1592,	<code>\VECTORADD</code> ..	1778–1780, 1783–
1594, 1627, 1647,		<a href="#">1016</a> , 1031,	1785, 1789–1791,
1663, 1667, 1682,		1034, 1198, 1199,	1795–1797, 1800–1802
1687, 1688, 1691,		1202–1204, 1263, 1265	<b>Z</b>
1701, 1704, 1708,		<code>\VECTORCOPY</code> . . .	<code>\ZEROfunction</code> . . . . .
1709, 1711, 1712,		<a href="#">946</a> , 970, 974	<a href="#">1697</a>
1715, 1718, 1721,		<code>\VECTORfunction</code> . . . . .	
1722, 1725, 1726,		<a href="#">1914</a>	
1728, 1729, 1733,		<code>\VECTORGLOBCOPY</code> . . . . .	
1734, 1738, 1739,		. . . . .	
1741, 1742, 1744,		. . . . .	
1745, 1749, 1750,		. . . . .	
		<a href="#">957</a> , 969, 973	
		<code>\VECTORMATRIXPRODUCT</code> . . .	
		. . . . .	
		<a href="#">1256</a> , 1294–1296	
		<code>\VECTORNORM</code> . . . . .	
		<a href="#">1066</a> ,	
		1072, 1077, 1078, 1426	
		<code>\VECTORPRODUCT</code> . . .	
		<a href="#">999</a> , 1015	