

# 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

2014/02/20

### 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).

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>I</b>	<b>The <code>calculator</code> package</b>	<b>5</b>
<b>2</b>	<b>Predefined numbers</b>	<b>5</b>
<b>3</b>	<b>Operations with numbers</b>	<b>5</b>
3.1	Assignments and comparisons . . . . .	5
3.2	Real arithmetic . . . . .	6

---

\*This document corresponds to `calculator` v.2.0 and `calculus` v.2.0, dated 2014/02/20.

3.2.1	The four basic operations . . . . .	6
3.2.2	Powers with integer exponent . . . . .	7
3.2.3	Absolute value, integer part and fractional part . . . . .	7
3.2.4	Truncation and rounding . . . . .	8
3.3	Integers . . . . .	8
3.3.1	Integer division, quotient and remainder . . . . .	9
3.3.2	Greatest common divisor and least common multiple . . . . .	10
3.3.3	Simplifying fractions . . . . .	10
3.4	Elementary functions . . . . .	10
3.4.1	Square roots . . . . .	10
3.4.2	Exponential and logarithm . . . . .	10
3.4.3	Trigonometric functions . . . . .	11
3.4.4	Hyperbolic functions . . . . .	13
3.4.5	Inverse trigonometric functions ( <i>new in version 2.0</i> ) . . . . .	14
3.4.6	Inverse hyperbolic functions ( <i>new in version 2.0</i> ) . . . . .	14
<b>4</b>	<b>Operations with lengths</b>	<b>15</b>
<b>5</b>	<b>Matrix arithmetic</b>	<b>15</b>
5.1	Vector operations . . . . .	16
5.1.1	Assignments . . . . .	16
5.1.2	Vector addition and subtraction . . . . .	16
5.1.3	Scalar-vector product . . . . .	16
5.1.4	Scalar (dot) product and euclidean norm . . . . .	17
5.1.5	Vector (cross) product ( <i>new in version 2.0</i> ) . . . . .	17
5.1.6	Unit vector parallel to a given vector (normalized vector) . . . . .	17
5.1.7	Absolute value (in each entry of a given vector) . . . . .	18
5.1.8	Angle between two vectors ( <i>new in version 2.0</i> ) . . . . .	18
5.2	Matrix operations . . . . .	18
5.2.1	Assignments . . . . .	18
5.2.2	Transposed matrix . . . . .	19
5.2.3	Matrix addition and subtraction . . . . .	19
5.2.4	Scalar-matrix product . . . . .	20
5.2.5	Matrix-vector product . . . . .	20
5.2.6	Product of two square matrices . . . . .	20
5.2.7	Determinant . . . . .	21
5.2.8	Inverse matrix . . . . .	21
5.2.9	Absolute value (in each entry) . . . . .	22
5.2.10	Solving a linear system . . . . .	22
<b>II</b>	<b>The calculus package</b>	<b>22</b>
<b>6</b>	<b>What is a <i>function</i>?</b>	<b>23</b>
<b>7</b>	<b>Predefined functions</b>	<b>23</b>

<b>8</b>	<b>Operations with functions</b>	<b>24</b>
<b>9</b>	<b>Polynomial functions</b>	<b>26</b>
<b>10</b>	<b>Vector-valued functions (or parametrically defined curves)</b>	<b>27</b>
<b>11</b>	<b>Vector-valued functions in polar coordinates</b>	<b>28</b>
<b>12</b>	<b>Low-level instructions</b>	<b>28</b>
12.1	The <code>\newfunction</code> declaration and its variants . . . . .	28
12.2	Vector functions and polar coordinates . . . . .	29
<b>III Implementation</b>		<b>30</b>
<b>13</b>	<b>calculator</b>	<b>30</b>
13.1	Internal lengths and special numbers . . . . .	30
13.2	Warning messages . . . . .	31
13.3	Operations with numbers . . . . .	33
13.4	Matrix arithmetics . . . . .	56
<b>14</b>	<b>calculus</b>	<b>69</b>
14.1	Error and info messages . . . . .	69
14.2	New functions . . . . .	71
14.3	Polynomials . . . . .	73
14.4	Elementary functions . . . . .	76
14.5	Operations with functions . . . . .	79

## 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.

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>1</sup> The new commands defined in this way work in any  $\text{\LaTeX}$  mode.

By example, this instruction

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

---

<sup>1</sup>Logically, the control sequences that represent special numbers (as `\numberPI`) does not need any argument.

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.

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 $$:
\[5^2=\sol\]
\begin{center}
\SQUAREROOT{5}\sol
Now, the \texttt{\textbackslash sol}
command is the square root of $$:
\[\sqrt{5}=\sol\]
\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>\numberHALFPPI</code>	$1.57079 \approx \pi/2$
<code>\numberTHREEHALFPPI</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

-1.256

```
\COPY{-1.256}{\sol}
\sol
```

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

Ex. 4

```
\MAX{1.256}{3.214}{\sol}
\[\max(1.256,3.214)=\sol\]
```

$$\max(1.256, 3.214) = 3.214$$

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

Ex. 5

```
\MIN{1.256}{3.214}{\sol}
\sol
```

$$1.256$$

## 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

```
\ADD{1.256}{3.214}{\sol}
$1.256+3.214=\sol$
```

$$1.256 + 3.214 = 4.47$$

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

Ex. 7

```
\SUBTRACT{1.256}{3.214}{\sol}
$1.256-3.214=\sol$
```

$$1.256 - 3.214 = -1.95801$$

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

Ex. 8

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

$$1.256 \times 3.214 = 4.03677$$

`\DIVIDE{<num1>}{<num2>}{<cmd>}` Quotient `num1 / num2`.<sup>2</sup>

Ex. 9

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

$$1.256/3.214 = 0.39078$$

---

<sup>2</sup>This command uses a modified version of the division algorithm of Claudio Beccari.

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

### 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}
$\left|vert-1.256\right|=\sol$
```

`\INTEGERPART{<num>}{<cmd>}` Integer part of *num*.<sup>3</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 $\solb$.
```

---

<sup>3</sup>The integer part of *x* is the largest integer that is less than or equal to *x*.

`\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*.

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.

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

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 `TeX` does not have this restriction in its integer arithmetic, the largest integer that can be used is 16383.

<sup>4</sup>Note that `\TRUNCATE[0]` is equivalent to `\INTEGERPART` only for non-negative numbers.



### 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>5</sup>

Ex. 19

$$\begin{aligned}
 435 &= 27 \times 16 + 3 \\
 27 &= 435 \times 0 + 27 \\
 -435 &= 27 \times (-17) + 24 \\
 435 &= -27 \times (-16) + 3 \\
 -435 &= -27 \times 17 + 24
 \end{aligned}$$

```
\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.

Ex. 20

$$\begin{aligned}
 &16 \\
 &0 \\
 &-17
 \end{aligned}$$

```
\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

$$\begin{aligned}
 435 &\equiv 3 \pmod{27} \\
 -435 &\equiv 24 \pmod{27}
 \end{aligned}$$

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

<sup>5</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.

### 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

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

```
\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 *num1*/*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>6</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}
$\log 0.5=\sol$
```

`\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 `\DEGREESIN`, `\DEGREESCOS`, ... commands.

`\SIN {⟨num⟩}{⟨cmd⟩}` Sine of *num*.

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

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

---

<sup>6</sup>9.704 is the logarithm of 16383, the largest number that supports the  $\text{T}_{\text{E}}\text{X}$ 's arithmetic.

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

Ex. 30

$\sin \pi/3 = 0.86601$   
 $\cos \pi/3 = 0.5$   
 $\tan \pi/3 = 1.73201$   
 $\cot \pi/3 = 0.57736$

`\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

$\sin 60^\circ = 0.86601$   
 $\cos 60^\circ = 0.49998$   
 $\tan 60^\circ = 1.73201$   
 $\cot 60^\circ = 0.57736$

`\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>}` arcoth 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>7</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>7</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( $\langle x1,y1 \rangle$ )( $\langle x2,y2 \rangle$ ){\cmd}`

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

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

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

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

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  $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & 22 \end{bmatrix}$  in  $\text{cmd11}$ ,  $\text{cmd12}$ ,  $\text{cmd21}$ ,  $\text{cmd22}$ .

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, \solf;`  
`\solg, \solh, \soli)`  
`$(\begin{bmatrix}`  
`\sola & \solb & \solc \\\`  
`\sold & \sole & \solf \\\`  
`\solg & \solh & \soli`  
`\end{bmatrix})$`

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

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>8</sup>

<sup>8</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>9</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>9</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 `TEX` 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

$$f(t) = 3t^2 - 2e^{-t} \cos t$$

then

$$f(5) = 74.99619$$

$$f'(5) = 29.99084$$

```
% exp(-t)
\SCALEVARIABLEfunction
{-1}{\EXPfunction}
{\NEGEXPfunction}

% exp(-t)cos(t)
\PRODUCTfunction
{\NEGEXPfunction}
{\COSfunction}
{\NEGEXPCOSfunction}

% 3t^2-2exp(-t)cos(t)
\LINEARCOMBINATIONfunction
{3}{\SQUAREfunction}
{-2}{\NEGEXPCOSfunction}
{\myfunction}

\myfunction{5}{\sol}{\Dsol}

If
\[
f(t)=3t^2-2\mathrm{e}^{-t}\cos t
\]
then
\[
\begin{gathered}
f(5)=\sol\\
f'(5)=\Dsol
\end{gathered}
\]
\]
```

## 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>}{<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}[2014/02/20 v.2.0]
```

#### 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@Warndivzero#1#2{%
10     \PackageWarning{calculator}%
11     {Division by 0.\MessageBreak
12     I can't define #1/#2}}
13
14 \def\cctr@Warnnogcd{%
15     \PackageWarning{calculator}%
16     {gcd(0,0) is not well defined}}
17
18 \def\cctr@Warnnuposrad#1{%
19     \PackageWarning{calculator}%
20     {The argument in square root\MessageBreak
21     must be non negative\MessageBreak
22     I can't define sqrt(#1)}}
23
24 \def\cctr@Warnnointexp#1#2{%
25     \PackageWarning{calculator}%
26     {The exponent in power function\MessageBreak
27     must be an integer\MessageBreak
28     I can't define #1^#2}}
29
30 \def\cctr@Warnbigarcsin#1{%
31     \PackageWarning{calculator}%
32     {The argument in arcsin\MessageBreak
33     must be a number between -1 and 1\MessageBreak
34     I can't define arcsin(#1)}}
35
36 \def\cctr@Warnbigarccos#1{%
37     \PackageWarning{calculator}%
38     {The argument in arccos\MessageBreak
39     must be a number between -1 and 1\MessageBreak
40     I can't define arccos(#1)}}
41
42 \def\cctr@Warnsmallarcosh#1{%
43     \PackageWarning{calculator}%
44     {The argument in arcosh\MessageBreak
45     must be a number greater or equal than 1\MessageBreak
46     I can't define arcosh(#1)}}
47
48 \def\cctr@Warnbigartanh#1{%
49     \PackageWarning{calculator}%
50     {The argument in artanh\MessageBreak
51     must be a number between -1 and 1\MessageBreak
52     I can't define artanh(#1)}}
53
54 \def\cctr@Warnsmallarcoth#1{%
55     \PackageWarning{calculator}%
56     {The argument in arcoth\MessageBreak
57     must be a number greater than 1\MessageBreak
```

```

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

```



```

108             The base\MessageBreak
109             in the exponential function must be positive.
110             \MessageBreak
111             I can't define #1^(#2)}}
112
113 \def\cctr@Warninflog#1{%
114     \PackageWarning{calculator}{%
115         The value of the variable\MessageBreak
116         in the logarithm function must be positive\MessageBreak
117         I can't define log(#1)}}
118
119 \def\cctr@Warncrossprod(#1)(#2){%
120     \PackageWarning{calculator}{%
121         {Vector product only defined\MessageBreak
122         for 3 dimmensional vectors.\MessageBreak
123         I can't define (#1)x(#2)}}
124
125 \def\cctr@Warnnoangle(#1)(#2){%
126     \PackageWarning{calculator}{%
127         {Angle between two vectors only defined\MessageBreak
128         for nonzero vectors.\MessageBreak
129         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`.

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

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

```
131 \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.

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

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

```
133 \def\@OUTPUTSOLS#1#2{\GLOBALCOPY{#1}{\cctr@outa}
134     \GLOBALCOPY{#2}{\cctr@outb}\endgroup
135     \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`.

```
136 \def\MAX#1#2#3{%
137     \ifdim #1\p@ < #2\p@
138     \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`.

```

139 \def\MIN#1#2#3{%
140   \ifdim #1\p@ > #2\p@
141     \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`.

```

142 \def\ABSVALUE#1#2{%
143   \ifdim #1\p@<\z@
144     \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`.

```

145 \def\MULTIPLY#1#2#3{\cctr@lengtha=#1\p@
146   \cctr@lengtha=#2\cctr@lengtha
147   \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`.

```

148 \def\ADD#1#2#3{\cctr@lengtha=#1\p@
149   \cctr@lengthb=#2\p@
150   \advance\cctr@lengtha by \cctr@lengthb
151   \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`.

```

152 \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`.

```

153 \def\DIVIDE#1#2#3{%
154   \begingroup
   Absolute values of dividend and divisor
155   \ABSVALUE{#1}{\cctr@tempD}
156   \ABSVALUE{#2}{\cctr@tempD}
   The sign of quotient
157   \ifdim#1\p@<\z@\ifdim#2\p@>\z@\COPY{-1}{\cctr@sign}
158     \else\COPY{1}{\cctr@sign}\fi
159   \else\ifdim#2\p@>\z@\COPY{1}{\cctr@sign}
160     \else\COPY{-1}{\cctr@sign}\fi
161   \fi}

```

Integer part of quotient

```
162 \@DIVIDE{\cctr@tempD}{\cctr@tempd}{\cctr@tempq}{\cctr@tempR}
163 \COPY{\cctr@tempq.}{\cctr@Q}
```

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

```
164 \COPY{0}{\cctr@ndec}
165 \@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.

```
166 \ifdim\cctr@tempR\p@<1638\p@
167 \MULTIPLY{\cctr@tempR}{10}{\cctr@tempD}
168 \else
169 \COPY{\cctr@tempR}{\cctr@tempD}
170 \MULTIPLY{\cctr@tempD}{0.1}{\cctr@tempD}
171 \fi
172 \@DIVIDE{\cctr@tempD}{\cctr@tempd}{\cctr@tempq}{\cctr@tempR}
173 \COPY{\cctr@Q\cctr@tempq.}{\cctr@Q}
174 \ADD{1}{\cctr@ndec}{\cctr@ndec}}%
```

Adjust the sign and return the solution.

```
175 \MULTIPLY{\cctr@sign}{\cctr@Q}{#3}
176 \@OUTPUTSOL{#3}
```

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

```
177 \def\@DIVIDE#1#2#3#4{%
178 \@INTEGERDIVIDE{#1}{#2}{#3}
179 \MULTIPLY{#2}{#3}{#4}
180 \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).

```
181 \def\@INTEGERDIVIDE#1#2#3{%
182 \cctr@lengtha=#1\p@
183 \cctr@lengthb=#2\p@
184 \ifdim\cctr@lengthb=\z@
185 \let#3\undefined
186 \cctr@Warndivzero#1#2%
187 \else
188 \divide\cctr@lengtha\cctr@lengthb
189 \COPY{\number\cctr@lengtha}{#3}
190 \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).

```
191 \def\LENGTHADD#1#2#3{\cctr@lengtha=#1
```

```

192     \ctr@lengthb=#2
193     \advance\ctr@lengtha by \ctr@lengthb
194     \setlength{#3}{\ctr@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).

```

195 \def\LENGTHSUBTRACT#1#2#3{%
196     \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*.

```

197 \def\LENGTHDIVIDE#1#2#3{%
198     \begingroup
199     \ctr@lengtha=#1
200     \ctr@lengthb=#2
201     \edef\ctr@tempa{\expandafter\strip@pt\ctr@lengtha}%
202     \edef\ctr@tempb{\expandafter\strip@pt\ctr@lengthb}%
203     \DIVIDE{\ctr@tempa}{\ctr@tempb}{#3}
204     \@OUTPUTSOL{#3}}

```

## Powers

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

```

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

```

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

```

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

```

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

```

207 \def\POWER#1#2#3{%
208     \begingroup
209     \INTEGERPART{#2}{\ctr@tempexp}
210     \ifdim \ctr@tempexp\p@<#2\p@
211         \ctr@Warnnointexp{#1}{#2}
212         \let#3\undefined
213     \else

```

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

```

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

```

```

215 \def\@POWER#1#2#3{%
216     \begingroup
217     \ifdim #2\p@<\z@

```

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

```

218         \DIVIDE{1}{#1}{\ctr@tempb}
219         \MULTIPLY{-1}{#2}{\ctr@tempc}
220         \@POWER{\ctr@tempb}{\ctr@tempc}{#3}
221     \else

```

```

222         \COPY{0}{\cctr@tempa}
223         \COPY{1}{#3}
224         \@whilenum \cctr@tempa<#2 \do {%
225             \MULTIPLY{#1}{#3}{#3}
226             \ADD{1}{\cctr@tempa}{\cctr@tempa}}%
227     \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.

```

228 \def\INTEGERDIVISION#1#2#3#4{%
229     \begingroup
230     \ABSVALUE{#2}{\cctr@tempd}
231     \@DIVIDE{#1}{#2}{#3}{#4}
232     \ifdim #4\p@<\z@
233         \ifdim #1\p@<\z@
234             \ifdim #2\p@<\z@
235                 \ADD{#3}{1}{#3}
236             \else
237                 \SUBTRACT{#3}{1}{#3}
238             \fi
239             \ADD{#4}{\cctr@tempd}{#4}
240     \fi\fi\@OUTPUTSOLS{#3}{#4}}

```

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

```

241 \def\MODULO#1#2#3{%
242     \begingroup
243     \INTEGERDIVISION{#1}{#2}{\cctr@temp}{#3}\@OUTPUTSOL{#3}}

```

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

```

244 \def\INTEGERQUOTIENT#1#2#3{%
245     \begingroup
246     \INTEGERDIVISION{#1}{#2}{#3}{\cctr@temp}\@OUTPUTSOL{#3}}

```

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

```

247 \def\@INTEGERPART#1.#2.#3)#4{\ifnum #11=1 \COPY{0}{#4}
248     \else \COPY{#1}{#4}\fi}
249 \def\@INTEGERPART#1#2{\expandafter\@INTEGERPART#1..}{#2}}
250 \def\INTEGERPART#1#2{\begingroup
251     \ifdim #1\p@<\z@
252         \MULTIPLY{-1}{#1}{\cctr@temp}
253         \INTEGERPART{\cctr@temp}{#2}
254     \ifdim #2\p@<\cctr@temp\p@
255         \SUBTRACT{-#2}{1}{#2}
256     \else \COPY{-#2}{#2}
257     \fi
258     \else
259         \@INTEGERPART{#1}{#2}
260     \fi\@OUTPUTSOL{#2}}

```

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

```
261 \let\FLOOR\INTEGERPART
```

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

```
262 \def\@FRACTIONALPART#1.#2.#3)#4{\ifnum #2=11 \COPY{0}{#4}
263 \else \COPY{0.#2}{#4}\fi}
264 \def\FRACTIONALPART#1#2{\expandafter\@FRACTIONALPART#1..}{#2}
265 \def\FRACTIONALPART#1#2{\begingroup
266 \ifdim #1\p@<\z@
267 \INTEGERPART{#1}{\cctr@tempA}
268 \SUBTRACT{#1}{\cctr@tempA}{#2}
269 \else
270 \@FRACTIONALPART{#1}{#2}
271 \fi\@OUTPUTSOL{#2}}
```

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

```
272 \def\TRUNCATE{\@ifnextchar[\@TRUNCATE\@TRUNCATE}
273 \def\@TRUNCATE#1#2{\@TRUNCATE[2]{#1}{#2}}
274 \def\@TRUNCATE[#1]#2#3{%
275 \begingroup
276 \INTEGERPART{#2}{\cctr@tempa}
277 \ifdim \cctr@tempa\p@ = #2\p@
278 \expandafter\@TRUNCATE#2.00000)[#1]{#3}
279 \else
280 \expandafter\@TRUNCATE#200000.)[#1]{#3}
281 \fi
282 \@OUTPUTSOL{#3}}
283 \def\@TRUNCATE#1.#2#3#4#5#6.#7)[#8]#9{%
284 \ifcase #8
285 \COPY{#1}{#9}
286 \or\COPY{#1.#2}{#9}
287 \or\COPY{#1.#2#3}{#9}
288 \or\COPY{#1.#2#3#4}{#9}
289 \or\COPY{#1.#2#3#4#5}{#9}
290 \fi}
```

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

```
291 \def\ROUND{\@ifnextchar[\@ROUND\@ROUND}
292 \def\@ROUND#1#2{\@ROUND[2]{#1}{#2}}
293 \def\@ROUND[#1]#2#3{%
294 \begingroup
295 \ifdim#2\p@<\z@
296 \MULTIPLY{-1}{#2}{\cctr@temp}
297 \@ROUND[#1]{\cctr@temp}{#3}\COPY{-#3}{#3}
298 \else
299 \@TRUNCATE[#1]{#2}{\cctr@temp}
300 \SUBTRACT{#2}{\cctr@temp}{\cctr@temp}
301 \POWER{10}{#1}{\cctr@tempb}
302 \MULTIPLY{\cctr@tempb}{\cctr@tempc}{\cctr@tempc}
303 \ifdim\cctr@tempc\p@<0.5\p@
```

```

304         \else
305             \DIVIDE{1}{\cctr@tempb}{\cctr@tempb}
306             \ADD{\cctr@tempe}{\cctr@tempb}{\cctr@tempe}
307         \fi
308         \@TRUNCATE[#1]{\cctr@tempe}{#3}
309     \fi
310     \@OUTPUTSOL{#3}}

```

`\GCD`  $\text{\GCD}\langle\#1\rangle\langle\#2\rangle\langle\#3\rangle$  Greatest common divisor, using the Euclidean algorithm

```

311 \def\GCD#1#2#3{%
312     \begingroup
313     \ABSVALUE{#1}{\cctr@tempa}
314     \ABSVALUE{#2}{\cctr@tempb}
315     \MAX{\cctr@tempa}{\cctr@tempb}{\cctr@tempc}
316     \MIN{\cctr@tempa}{\cctr@tempb}{\cctr@tempa}
317     \COPY{\cctr@tempc}{\cctr@tempb}
318     \ifnum \cctr@tempa = 0
319         \ifnum \cctr@tempb = 0
320             \cctr@Warnnogcd
321             \let#3\undefined
322         \else
323             \COPY{\cctr@tempb}{#3}
324         \fi
325     \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.

```

326         \@whilenum \cctr@tempa > \z@ \do {%
327             \COPY{\cctr@tempa}{#3}%
328             \MODULO{\cctr@tempb}{\cctr@tempa}{\cctr@tempc}%
329             \COPY\cctr@tempa\cctr@tempb%
330             \COPY\cctr@tempc\cctr@tempa}
331     \fi\ignorespaces\@OUTPUTSOL{#3}}

```

`\LCM`  $\text{\LCM}\langle\#1\rangle\langle\#2\rangle\langle\#3\rangle$  Least common multiple.

```

332 \def\LCM#1#2#3{%
333     \GCD{#1}{#2}{#3}%
334     \ifx #3\undefined \COPY{0}{#3}
335     \else
336         \DIVIDE{#1}{#3}{#3}
337         \MULTIPLY{#2}{#3}{#3}
338         \ABSVALUE{#3}{#3}
339     \fi}

```

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

```

340 \def\FRACTIONSIMPLIFY#1#2#3#4{%
341     \ifnum #1=\z@
342         \COPY{0}{#3}\COPY{1}{#4}
343     \else

```

```

344      \GCD{#1}{#2}{#3}%
345      \DIVIDE{#2}{#3}{#4}
346      \DIVIDE{#1}{#3}{#3}
347      \ifnum #4<0 \MULTIPLY{-1}{#4}{#4}\MULTIPLY{-1}{#3}{#3}\fi
348      \fi\ignorespaces}

```

## Elementary functions

### Square roots

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

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

```

349 \def\SQUAREROOT#1#2{%
350     \begingroup
351     \ifdim #1\p@ = \z@
352         \COPY{0}{#2}
353     \else
354         \ifdim #1\p@ < \z@
355             \let#2\undefined
356             \cctr@Warnnuposrad{#1}%
357         \else

```

We take `#1` as the initial approximation.

```

358     \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.

```

359     \cctr@lengthb=5\p@

```

Successive iterations

```

360     \@whilenum \cctr@lengthb>\cctr@epsilon \do {%

```

Copy the actual approximation to `\cctr@tempw`

```

361         \COPY{#2}{\cctr@tempw}
362         \DIVIDE{#1}{\cctr@tempw}{\cctr@tempz}
363         \ADD{\cctr@tempw}{\cctr@tempz}{\cctr@tempz}
364         \DIVIDE{\cctr@tempz}{2}{\cctr@tempz}

```

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

```

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

```

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

```

366         \SUBTRACT{#2}{\cctr@tempw}{\cctr@tempz}
367         \cctr@lengthb=\cctr@tempw\p@%
368         \ifnum
369             \cctr@lengthb<\z@ \cctr@lengthb=-\cctr@lengthb
370         \fi}
371     \fi\fi\@OUTPUTSOL{#2}}

```

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

```

372 \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.
373 \def\SIN#1#2{%
374 \begingroup
    Exact sine for  $t \in \{\pi/2, -\pi/2, 3\pi/2\}$ 
375 \ifdim #1\p@=-\numberHALFPI\p@ \COPY{-1}{#2}
376 \else
377 \ifdim #1\p@=\numberHALFPI\p@ \COPY{1}{#2}
378 \else
379 \ifdim #1\p@=\numberTHREEHALFPI\p@ \COPY{-1}{#2}
380 \else
    If  $|t| > \pi/2$ , change  $t$  to a smaller value.
381 \ifdim#1\p@<-\numberHALFPI\p@
382 \ADD{#1}{\numberTWOPI}{\cctr@tempb}
383 \SIN{\cctr@tempb}{#2}
384 \else
385 \ifdim #1\p@<\numberHALFPI\p@
    Compute the sine.
386 \@BASICSINE{#1}{#2}
387 \else
388 \ifdim #1\p@<\numberTHREEHALFPI\p@
389 \SUBTRACT{\numberPI}{#1}{\cctr@tempb}
390 \SIN{\cctr@tempb}{#2}
391 \else
392 \SUBTRACT{#1}{\numberTWOPI}{\cctr@tempb}
393 \SIN{\cctr@tempb}{#2}
394 \fi\fi\fi\fi\fi\fi\@OUTPUTSOL{#2}}

```

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

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

```

395 \def\@BASICSINE#1#2{%
396 \begingroup
397 \ABSVALUE{#1}{\cctr@tempa}
    Exact sine of zero
398 \ifdim\cctr@tempa\p@=\z@ \COPY{0}{#2}
399 \else
    For  $t$  very close to zero,  $\sin t \approx t$ .
400 \ifdim \cctr@tempa\p@<0.009\p@\COPY{#1}{#2}
401 \else

```

Compute the continued fraction.

```

402          \SQUARE{#1}{\cctr@tempa}
403          \DIVIDE{\cctr@tempa}{42}{#2}
404          \SUBTRACT{1}{#2}{#2}
405          \MULTIPLY{#2}{\cctr@tempa}{#2}
406          \DIVIDE{#2}{20}{#2}
407          \SUBTRACT{1}{#2}{#2}
408          \MULTIPLY{#2}{\cctr@tempa}{#2}
409          \DIVIDE{#2}{6}{#2}
410          \SUBTRACT{1}{#2}{#2}
411          \MULTIPLY{#2}{#1}{#2}
412          \fi\fi\@OUTPUTSOL{#2}

```

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

```

413 \def\COS#1#2{%
414     \begingroup
415     \ADD{\numberHALFPI}{#1}{\cctr@tempc}
416     \SIN{\cctr@tempc}{#2}\@OUTPUTSOL{#2}

```

`\TAN` `\TAN{<#1>}{<#2>}`. Tangent of *#1*.

```

417 \def\TAN#1#2{%
418     \begingroup
419     Tangent is infinite for  $t = \pm\pi/2$ 
420     \ifdim #1\p@=-\numberHALFPI\p@
421         \cctr@WarninfTan{#1}
422         \let#2\undefined
423     \else
424         \ifdim #1\p@=\numberHALFPI\p@
425             \cctr@WarninfTan{#1}
426             \let#2\undefined
427         \else

```

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

```

427         \ifdim #1\p@<-\numberHALFPI\p@
428             \ADD{#1}{\numberPI}{\cctr@tempb}
429             \TAN{\cctr@tempb}{#2}
430         \else
431             \ifdim #1\p@<\numberHALFPI\p@

```

Compute the tangent.

```

432             \@BASICTAN{#1}{#2}
433         \else
434             \SUBTRACT{#1}{\numberPI}{\cctr@tempb}
435             \TAN{\cctr@tempb}{#2}
436         \fi\fi\fi\@OUTPUTSOL{#2}

```

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

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

```

437 \def\@BASICTAN#1#2{%
438     \begingroup
439     \ABSVALUE{#1}{\cctr@tempa}
Exact tangent of zero.
440     \ifdim\cctr@tempa\p@=\z@ \COPY{0}{#2}
441     \else
For  $t$  very close to zero,  $\tan t \approx t$ .
442     \ifdim\cctr@tempa\p@<0.04\p@
443     \COPY{#1}{#2}
444     \else
Compute the continued fraction.
445     \DIVIDE{#1}{11}{#2}
446     \DIVIDE{9}{#1}{\cctr@tempa}
447     \SUBTRACT{\cctr@tempa}{#2}{#2}
448     \DIVIDE{1}{#2}{#2}
449     \DIVIDE{7}{#1}{\cctr@tempa}
450     \SUBTRACT{\cctr@tempa}{#2}{#2}
451     \DIVIDE{1}{#2}{#2}
452     \DIVIDE{5}{#1}{\cctr@tempa}
453     \SUBTRACT{\cctr@tempa}{#2}{#2}
454     \DIVIDE{1}{#2}{#2}
455     \DIVIDE{3}{#1}{\cctr@tempa}
456     \SUBTRACT{\cctr@tempa}{#2}{#2}
457     \DIVIDE{1}{#2}{#2}
458     \DIVIDE{1}{#1}{\cctr@tempa}
459     \SUBTRACT{\cctr@tempa}{#2}{#2}
460     \DIVIDE{1}{#2}{#2}
461     \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,  $\cot t = 1/\tan t$ .

```

462 \def\COT#1#2{%
463     \begingroup
464     \COS{#1}{#2}
465     \ifdim #2\p@ = \z@
466     \COPY{0}{#2}
467     \else

```

```

468     \TAN{#1}{#2}
469     \ifdim #2\p@ = \z@
470     \ctr@Warninfcotan{#1}
471     \let#2\undefined
472     \else
473     \DIVIDE{1}{#2}{#2}
474     \fi\fi\@OUTPUTSOL{#2}}

```

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

```
475 \def\DEGtoRAD#1#2{\DIVIDE{#1}{57.29578}{#2}}
```

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

```
476 \def\radtoDEG#1#2{\MULTIPLY{#1}{57.29578}{#2}}
```

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

```

477 \def\REDUCERADIANSANGLE#1#2{%
478     \COPY{#1}{#2}
479     \ifdim #1\p@ < -\numberPI\p@
480         \ADD{#1}{\numberTWOPI}{#2}
481         \REDUCERADIANSANGLE{#2}{#2}
482     \fi
483     \ifdim #1\p@ > \numberPI\p@
484         \SUBTRACT{#1}{\numberTWOPI}{#2}
485         \REDUCERADIANSANGLE{#2}{#2}
486     \fi
487     \ifdim #1\p@ = -180\p@ \COPY{\numberPI}{#2} \fi}

```

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

```

488 \def\REDUCEDEGREESANGLE#1#2{%
489     \COPY{#1}{#2}
490     \ifdim #1\p@ < -180\p@
491         \ADD{#1}{360}{#2}
492         \REDUCEDEGREESANGLE{#2}{#2}
493     \fi
494     \ifdim #1\p@ > 180\p@
495         \SUBTRACT{#1}{360}{#2}
496         \REDUCEDEGREESANGLE{#2}{#2}
497     \fi
498     \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*.

```
499 \def\DEGREESSIN{\@ifnextchar [\@@DEGREESSIN\@DEGREESSIN}
```

`\DEGREESCOS` `\DEGREESCOS[<#1>]{<#2>}{<#3>}`. Cosine of *#2 degrees*.

```
500 \def\DEGREESCOS{\@ifnextchar [\@@DEGREESCOS\@DEGREESCOS}
```

```

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

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

\@DEGREESSIN \@DEGREESSIN computes the sine in sexagesimal degrees.
503 \def\@DEGREESSIN#1#2{%
504   \begingroup
505   \ifdim #1\p@=-90\p@ \COPY{-1}{#2}
506   \else
507     \ifdim #1\p@=90\p@ \COPY{1}{#2}
508     \else
509       \ifdim #1\p@=270\p@ \COPY{-1}{#2}
510       \else
511         \ifdim#1\p@<-90\p@
512           \ADD{#1}{360}{\ctr@tempb}
513           \DEGREESSIN{\ctr@tempb}{#2}
514         \else
515           \ifdim #1\p@<90\p@
516             \DEGtoRAD{#1}{\ctr@tempb}
517             \@BASICSINE{\ctr@tempb}{#2}
518           \else
519             \ifdim #1\p@<270\p@
520               \SUBTRACT{180}{#1}{\ctr@tempb}
521               \DEGREESSIN{\ctr@tempb}{#2}
522             \else
523               \SUBTRACT{#1}{360}{\ctr@tempb}
524               \DEGREESSIN{\ctr@tempb}{#2}
525           \fi\fi\fi\fi\fi\fi\@OUTPUTSOL{#2}}

\@DEGREESCOS \@DEGREESCOS computes the cosine in sexagesimal degrees.
526 \def\@DEGREESCOS#1#2{%
527   \begingroup
528   \ADD{90}{#1}{\ctr@tempc}
529   \DEGREESSIN{\ctr@tempc}{#2}\@OUTPUTSOL{#2}}

\@DEGREESTAN \@DEGREESTAN computes the tangent in sexagesimal degrees.
530 \def\@DEGREESTAN#1#2{%
531   \begingroup
532   \ifdim #1\p@=-90\p@
533     \ctr@Warninftan{#1}
534     \let#2\undefined
535   \else
536     \ifdim #1\p@=90\p@
537       \ctr@Warninftan{#1}
538       \let#2\undefined
539     \else
540       \ifdim #1\p@<-90\p@
541         \ADD{#1}{180}{\ctr@tempb} \DEGREESTAN{\ctr@tempb}{#2}

```

```

542         \else
543         \ifdim #1\p@<90\p@
544             \DEGtoRAD{#1}{\cctr@tempb}
545             \@BASICTAN{\cctr@tempb}{#2}
546         \else
547             \SUBTRACT{#1}{180}{\cctr@tempb}
548             \DEGREESTAN{\cctr@tempb}{#2}
549     \fi\fi\fi\fi\@OUTPUTSOL{#2}}

```

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

```

550 \def\@DEGREESCOT#1#2{%
551     \begingroup
552     \DEGREESCOS{#1}{#2}
553     \ifdim #2\p@ = \z@
554     \COPY{0}{#2}
555     \else
556     \DEGREESTAN{#1}{#2}
557     \ifdim #2\p@ = \z@
558     \cctr@Warninfcotan{#1}
559     \let#2\undefined
560     \else
561     \DIVIDE{1}{#2}{#2}
562     \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*.

```

563 \def\@@DEGREESSIN[#1]#2#3{\@CONVERTDEG{#1}{#2}
564     \@DEGREESSIN{\@DEGREES}{#3}}

```

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

```

565 \def\@@DEGREESCOS[#1]#2#3{\@CONVERTDEG{#1}{#2}
566     \DEGREESCOS{\@DEGREES}{#3}}

```

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

```

567 \def\@@DEGREESTAN[#1]#2#3{\@CONVERTDEG{#1}{#2}
568     \DEGREESTAN{\@DEGREES}{#3}}

```

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

```

569 \def\@@DEGREESCOT[#1]#2#3{\@CONVERTDEG{#1}{#2}
570     \DEGREESCOT{\@DEGREES}{#3}}

```

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

```

571 \def\@CONVERTDEG#1#2{\DIVIDE{#2}{#1}{\@DEGREES}
572     \MULTIPLY{\@DEGREES}{360}{\@DEGREES}}

```

## Exponential functions

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

```
573 \def\EXP{\@ifnextchar[\@EXP\@EXP}
```

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

```
574 \def\@EXP[#1]#2#3{%
```

```
575     \begingroup
```

$\#1$  must be a positive number.

```
576     \ifdim #1\p<\cctr@epsilon
```

```
577         \cctr@Warninfexpb{#1}{#2}
```

```
578         \let#3\undefined
```

```
579     \else
```

$a^b = \exp(b \log a)$ .

```
580         \LOG{#1}{\cctr@log}
```

```
581         \MULTIPLY{#2}{\cctr@log}{\cctr@log}
```

```
582         \EXP{\cctr@log}{#3}
```

```
583     \fi\@OUTPUTSOL{#3}}
```

`\@EXP \@EXP{⟨#1⟩}{⟨#2⟩}` computes  $\#3 = e^{\#2}$

```
584 \def\@EXP#1#2{%
```

```
585     \begingroup
```

```
586     \ABSVALUE{#1}{\cctr@absval}
```

If  $|t|$  is greater than `\cctr@logmaxnum` then  $\exp t$  is too large.

```
587     \ifdim \cctr@absval\p>\cctr@logmaxnum\p@
```

```
588         \cctr@Warninfexp{#1}
```

```
589         \let#2\undefined
```

```
590     \else
```

```
591         \ifdim #1\p@ < \z@
```

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

```
592         \ifdim #1\p@ > -6.00002\p@
```

```
593             \@BASICEXP{#1}{#2}
```

```
594         \else
```

```
595             \DIVIDE{#1}{2}{\cctr@expt}
```

```
596             \EXP{\cctr@expt}{\cctr@expy}
```

```
597             \SQUARE{\cctr@expy}{#2}
```

```
598         \fi
```

```
599     \else
```

```
600         \ifdim #1\p@ < 3.00002\p@
```

```
601             \@BASICEXP{#1}{#2}
```

```
602         \else
```

```
603             \DIVIDE{#1}{2}{\cctr@expt}
```

```
604             \EXP{\cctr@expt}{\cctr@expy}
```

```
605             \SQUARE{\cctr@expy}{#2}
```

```
606         \fi
```

```
607 \fi\fi\@OUTPUTSOL{#2}}
```

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

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

```

608 \def\@BASICEXP#1#2{%
609     \begingroup
610     \SQUARE{#1}\cctr@tempa
611     \DIVIDE{\cctr@tempa}{396}{#2}
612     \ADD{1}{#2}{#2}
613     \DIVIDE\cctr@tempa{#2}{#2}
614     \DIVIDE{#2}{256}{#2}
615     \ADD{1}{#2}{#2}
616     \DIVIDE\cctr@tempa{#2}{#2}
617     \DIVIDE{#2}{140}{#2}
618     \ADD{1}{#2}{#2}
619     \DIVIDE\cctr@tempa{#2}{#2}
620     \DIVIDE{#2}{60}{#2}
621     \ADD{1}{#2}{#2}
622     \DIVIDE\cctr@tempa{#2}{#2}
623     \DIVIDE{#2}{6}{#2}
624     \ADD{2}{#2}{#2}
625     \SUBTRACT{#2}{#1}{#2}
626     \DIVIDE{#1}{#2}{#2}
627     \MULTIPLY{2}{#2}{#2}
628     \ADD{1}{#2}{#2}\@OUTPUTSOL{#2}}

```

### Hyperbolic functions

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

```

629 \def\COSH#1#2{%
630     \begingroup
631     \ABSVALUE{#1}{\cctr@absval}
632     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
633         \cctr@Warninfexp{#1}
634         \let#2\undefined
635     \else
636         \EXP{#1}{\cctr@exp}
637         \MULTIPLY{-1}{#1}{\cctr@minust}
638         \EXP{\cctr@minust}{\cctr@expminusx}
639         \ADD{\cctr@exp}{\cctr@expminusx}{#2}
640         \DIVIDE{#2}{2}{#2}
641     \fi\@OUTPUTSOL{#2}}

```



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

```

642 \def\SINH#1#2{%
643     \begingroup
644     \ABSVALUE{#1}{\cctr@absval}
645     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
646         \cctr@Warninfexp{#1}
647         \let#2\undefined
648     \else
649         \EXP{#1}{\cctr@expx}
650         \MULTIPLY{-1}{#1}{\cctr@minust}
651         \EXP{\cctr@minust}{\cctr@expminux}
652         \SUBTRACT{\cctr@expx}{\cctr@expminux}{#2}
653         \DIVIDE{#2}{2}{#2}
654     \fi\@OUTPUTSOL{#2}}

```

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

```

655 \def\TANH#1#2{%
656     \begingroup
657     \ABSVALUE{#1}{\cctr@absval}
658     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
659         \cctr@Warninfexp{#1}
660         \let#2\undefined
661     \else
662         \SINH{#1}{\cctr@tanhnum}
663         \COSH{#1}{\cctr@tanhden}
664         \DIVIDE{\cctr@tanhnum}{\cctr@tanhden}{#2}
665     \fi\@OUTPUTSOL{#2}}

```

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

```

666 \def\COTH#1#2{%
667     \begingroup
668     \ABSVALUE{#1}{\cctr@absval}
669     \ifdim \cctr@absval\p@>\cctr@logmaxnum\p@
670         \cctr@Warninfexp{#1}
671         \let#2\undefined
672     \else
673         \SINH{#1}{\cctr@tanhden}
674         \COSH{#1}{\cctr@tanhnum}
675         \DIVIDE\cctr@tanhnum\cctr@tanhden{#2}
676     \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  $e$ .

```

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

```

`\@LOG` `\@LOG` $\{\langle\#1\rangle\}{\langle\#2\rangle}$  computes  $\#2 = \log \#1$

```

678 \def\@LOG#1#2{%
679     \begingroup

```

The argument  $t$  must be positive.

```

680     \ifdim #1\p@<\cctr@epsilon
681     \cctr@Warninflog{#1}
682     \let#2\undefined
683     \else
684     \ifdim #1\p@ > \numberETWO\p@
    If  $t > e^2$ ,  $\log t = \log e + \log(t/e) = 1 + \log(t/e)$ 
685     \DIVIDE{#1}{\numberE}{\cctr@ae}
686     \@LOG{\cctr@ae}{#2}
687     \ADD{1}{#2}{#2}
688     \else
689     \ifdim #1\p@ < 1\p@
    If  $t < 1$ ,  $\log t = \log(1/e) + \log(te) = -1 + \log(te)$ 
690     \MULTIPLY{\numberE}{#1}{\cctr@ae}
691     \LOG{\cctr@ae}{#2}
692     \SUBTRACT{#2}{1}{#2}
693     \else
    For  $t \in [1, e^2]$  we call \@BASICLOG.
694     \@BASICLOG{#1}{#2}
695 \fi\fi\fi\@OUTPUTSOL{#2}}

```

\@@LOG \@@LOG[ $\langle \#1 \rangle$ ]{ $\langle \#2 \rangle$ }{ $\langle \#3 \rangle$ } computes  $\#3 = \log_{\#1} \#2 = \log(\#2)/\log(\#1)$

```

696 \def\@@LOG[#1]#2#3{\begingroup
697     \@LOG{#1}{\cctr@loga}
698     \@LOG{#2}{\cctr@logx}
699     \DIVIDE{\cctr@logx}{\cctr@loga}{#3}\@OUTPUTSOL{#3}}

```

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

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

```

700 \def\@BASICLOG#1#2{\begingroup
701 % We take $\textit{\langle \#1 \rangle}-1$ as the initial approximation.
702 % \begin{macrocode}
703     \SUBTRACT{#1}{1}{\cctr@tempw}
    We start with \cctr@lengthb=5\p@ to ensure almost one iteration.
704     \cctr@lengthb=5\p@%
    Successive iterations
705     \@whilenum \cctr@lengthb>\cctr@epsilon \do {%
706         \COPY{\cctr@tempw}{\cctr@tempoldw}
707         \EXP{\cctr@tempw}{\cctr@tempxw}
708         \DIVIDE{#1}{\cctr@tempxw}{\cctr@tempxw}
709         \ADD{\cctr@tempw}{\cctr@tempxw}{\cctr@tempw}
710         \SUBTRACT{\cctr@tempw}{1}{\cctr@tempw}
711         \SUBTRACT{\cctr@tempw}{\cctr@tempoldw}{\cctr@tempdif}
712         \cctr@lengthb=\cctr@tempdif\p@%

```

```

713         \ifnum
714             \cctr@lengthb<\z@ \cctr@lengthb=-\cctr@lengthb
715         \fi}%
716     \COPY{\cctr@tempw}{#2}\@OUTPUTSOL{#2}}

```

### Inverse trigonometric functions

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

```

717 \def\ARCSIN#1#2{%
718     \begingroup
719     \ifdim #1\p@ = \z@
720         \COPY{0}{#2}
721     \else
722         \ifdim #1\p@ = 1\p@
723             \COPY{\numberHALFPI}{#2}
724         \else
725             \ifdim #1\p@ = -1\p@
726                 \COPY{-\numberHALFPI}{#2}
727             \else
728                 \ifdim #1\p@ > 1\p@
729                     \let#2\undefined
730                 \cctr@Warnbigarcsin{#1}
731             \else
732                 \ifdim #1\p@ < -1\p@
733                     \let#2\undefined
734                 \cctr@Warnbigarcsin{#1}
735             \else

```

If  $x$  is close to 1 we use  $\arcsin x = \pi/2 - 2 \arcsin \sqrt{(1-x)/2}$

```

736         \ifdim #1\p@ > 0.89\p@
737             \SUBTRACT{1}{#1}{\cctr@tempx}
738             \DIVIDE{\cctr@tempx}{2}{\cctr@tempx}
739             \SQRT{\cctr@tempx}{\cctr@tempxx}
740             \ARCSIN{\cctr@tempxx}{#2}
741             \MULTIPLY{2}{#2}{#2}
742             \SUBTRACT{\numberHALFPI}{#2}{#2}
743         \else

```

Symmetrically, for  $x$  close to  $-1$ ,  $\arcsin x = -\pi/2 + 2 \arcsin \sqrt{(1+x)/2}$

```

744         \ifdim #1\p@ < -0.89\p@
745             \ADD{1}{#1}{\cctr@tempx}
746             \DIVIDE{\cctr@tempx}{2}{\cctr@tempx}
747             \SQRT{\cctr@tempx}{\cctr@tempxx}
748             \ARCSIN{\cctr@tempxx}{#2}
749             \MULTIPLY{2}{#2}{#2}
750             \SUBTRACT{#2}{\numberHALFPI}{#2}
751         \else

```

We take `#1` as the initial approximation.

```

752         \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

```
753             \ABSVALUE{#1}{\cctr@tempy}
754             \ifdim \cctr@tempy\p@ < 0.04\p@
755             \else
```

\cctr@lengthb will be the difference of two successive iterations, and \cctr@tempoldy, \cctr@tempy will be the two last iterations.

We start with \cctr@lengthb=5\p@ and \cctr@tempy=16383 to ensure almost one iteration.

```
756             \cctr@lengthb=5\p@
757             \COPY{16383}{\cctr@tempy}
```

Successive iterations

```
758             \@whilenum \cctr@lengthb>\cctr@epsilon \do {%
```

Copy the actual approximation to \cctr@tempw

```
759             \COPY{#2}{\cctr@tempw}
760             \COPY{\cctr@tempy}{\cctr@tempoldy}
761             \SIN{\cctr@tempw}{\cctr@tempz}
762             \SUBTRACT{\cctr@tempz}{#1}{\cctr@tempz}
763             \COS{\cctr@tempw}{\cctr@tempy}
764             \DIVIDE{\cctr@tempz}{\cctr@tempy}{\cctr@tempz}
765             \SUBTRACT{\cctr@tempw}{\cctr@tempz}{\cctr@tempz}
```

Now, \cctr@tempz is the new approximation.

```
766             \COPY{\cctr@tempz}{#2}
```

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

```
767             \SUBTRACT{#2}{\cctr@tempw}{\cctr@tempy}
768             \ABSVALUE{\cctr@tempy}{\cctr@tempy}
769             \cctr@lengthb=\cctr@tempy\p@%
770             \ifdim\cctr@tempy\p@=\cctr@tempoldy\p@
771             \cctr@lengthb=\z@
772             \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  $\arccos x = \pi/2 - \arcsin x$ .

```
773 \def\ARCCOS#1#2{%
774     \begingroup
775     \ifdim #1\p@ = \z@
776         \COPY{\numberHALFPI}{#2}
777     \else
778         \ifdim #1\p@ = 1\p@
779             \COPY{0}{#2}
780         \else
781             \ifdim #1\p@ = -1\p@
782                 \COPY{\numberPI}{#2}
783             \else
784                 \ifdim #1\p@ > 1\p@
785                     \let#2\undefined
786                 \cctr@Warnbigarccos{#1}
787             \else
```

```

788             \ifdim #1\p@ < -1\p@
789                 \let#2\undefined
790                 \cctr@Warnbigarccos{#1}
791             \else
792                 \ARCSIN{#1}{#2}
793                 \SUBTRACT{\numberHALFPI}{#2}{#2}
794             \fi\fi\fi\fi\fi\@OUTPUTSOL{#2}}

```

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

```

795 \def\ARCTAN#1#2{%
796     \begingroup
    If  $|t| > 1$ , compute arctan  $x$  using  $\arctan x = \operatorname{sign}(x)\pi/2 - \arctan(1/x)$ .
797         \ifdim#1\p@<-1\p@
798             \DIVIDE{1}{#1}{\cctr@tempb}
799             \ARCTAN{\cctr@tempb}{#2}
800             \SUBTRACT{-\numberHALFPI}{#2}{#2}
801         \else
802             \ifdim#1\p@>1\p@
803                 \DIVIDE{1}{#1}{\cctr@tempb}
804                 \ARCTAN{\cctr@tempb}{#2}
805                 \SUBTRACT{\numberHALFPI}{#2}{#2}
806             \else
    For  $-1 \leq x \leq 1$  call \@BASICARCTAN.
807                 \@BASICARCTAN{#1}{#2}
808             \fi
809             \fi\@OUTPUTSOL{#2}}

```

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

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

```

810 \def\@BASICARCTAN#1#2{%
811     \begingroup
    Exact arctan of zero
812         \ifdim#1\p@=\z@ \COPY{0}{#2}
813         \else
    Compute the continued fraction.
814             \SQUARE{#1}{\cctr@tempa}
815             \MULTIPLY{64}{\cctr@tempa}{#2}
816             \ADD{15}{#2}{#2}
817             \DIVIDE{\cctr@tempa}{#2}{#2}
818             \MULTIPLY{49}{#2}{#2}

```

```

819          \ADD{13}{#2}{#2}
820          \DIVIDE{\cctr@tempa}{#2}{#2}
821          \MULTIPLY{36}{#2}{#2}
822          \ADD{11}{#2}{#2}
823          \DIVIDE{\cctr@tempa}{#2}{#2}
824          \MULTIPLY{25}{#2}{#2}
825          \ADD{9}{#2}{#2}
826          \DIVIDE{\cctr@tempa}{#2}{#2}
827          \MULTIPLY{16}{#2}{#2}
828          \ADD{7}{#2}{#2}
829          \DIVIDE{\cctr@tempa}{#2}{#2}
830          \MULTIPLY{9}{#2}{#2}
831          \ADD{5}{#2}{#2}
832          \DIVIDE{\cctr@tempa}{#2}{#2}
833          \MULTIPLY{4}{#2}{#2}
834          \ADD{3}{#2}{#2}
835          \DIVIDE{\cctr@tempa}{#2}{#2}
836          \ADD{1}{#2}{#2}
837          \DIVIDE{#1}{#2}{#2}
838      \fi \@OUTPUTSOL{#2}}

```

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

```

839 \def\ARCCOT#1#2{%
840     \begingroup
841     \ARCTAN{#1}{#2}
842     \SUBTRACT{\numberHALFPI}{#2}{#2}
843     \@OUTPUTSOL{#2}}

```

### Inverse hyperbolic functions

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

```

844 \def\ARSINH#1#2{%
845     \begingroup
846     \SQUARE{#1}{\cctr@tempa}
847     \ADD{1}{\cctr@tempa}{\cctr@tempa}
848     \SQRT{\cctr@tempa}{\cctr@tempb}
849     \ADD{#1}{\cctr@tempb}{\cctr@tempb}
850     \LOG\cctr@tempb{#2}
851     \@OUTPUTSOL{#2}}

```

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

```

852 \def\ARCOSH#1#2{%
853     \begingroup
854     If  $x < 1$ , this function is no defined
855     \ifdim#1\p<1\p@
856     \let#2\undefined
857     \cctr@Warnsmallarcosh{#1}
858     \else

```

```

858      \SQUARE{#1}{\cctr@tempa}
859      \SUBTRACT{\cctr@tempa}{1}{\cctr@tempa}
860      \SQRT{\cctr@tempa}{\cctr@tempb}
861      \ADD{#1}{\cctr@tempb}{\cctr@tempb}
862      \LOG\cctr@tempb{#2}
863      \fi\@OUTPUTSOL{#2}}

```

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

```

864 \def\ARTANH#1#2{%
865   \begingroup
      If  $|x| \geq 1$ , this function is no defined
866     \ifdim#1\p@<-0.99998\p@
867       \let#2\undefined
868       \cctr@Warnbigartanh{#1}
869     \else
870       \ifdim#1\p@>0.99998\p@
871         \let#2\undefined
872         \cctr@Warnbigartanh{#1}
873       \else
874         \COPY{#1}{\cctr@tempa}
875         \ADD1\cctr@tempa\cctr@tempb
876         \SUBTRACT1\cctr@tempa\cctr@tempc
877         \LOG\cctr@tempb\cctr@tempB
878         \LOG\cctr@tempc\cctr@tempC
879         \SUBTRACT\cctr@tempB\cctr@tempC{#2}
880         \DIVIDE{#2}{2}{#2}
881       \fi
882     \fi\@OUTPUTSOL{#2}}

```

\ARCOTH \ARCOTH{<#1>}{<#2>}. Inverse hyperbolic cotangent of #1:  
 $\operatorname{arcoth} x = \operatorname{sign}(x) \frac{1}{2} \log((x+1) - \log(x-1))$

```

883 \def\ARCOTH#1#2{%
884   \begingroup
      If  $|x| \leq 1$ , this function is no defined
885     \ifdim#1\p@>-0.99998\p@
886     \ifdim#1\p@<0.99998\p@
887       \let#2\undefined
888       \cctr@Warnsmallarcoth{#1}
889     \else
890       \ifdim#1\p@>\p@
      For  $x > 1$ , calculate  $\operatorname{arcoth} x = \frac{1}{2} \log((x+1) - \log(x-1))$ 
891         \COPY{#1}{\cctr@tempa}
892         \ADD1\cctr@tempa\cctr@tempb
893         \SUBTRACT\cctr@tempa1\cctr@tempc
894         \LOG\cctr@tempb\cctr@tempB
895         \LOG\cctr@tempc\cctr@tempC
896         \SUBTRACT\cctr@tempB\cctr@tempC{#2}
897         \DIVIDE{#2}{2}{#2}

```

```

898         \else
899         \fi
900     \fi
901     \else
    For  $x < -1$ , calculate  $-\operatorname{artanh}(-x)$ 
902         \MULTIPLY{-1}{#1}{\cctr@tempa}
903         \ARCOATH{\cctr@tempa}{#2}
904         \COPY{-#2}{#2}
905     \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.

```

906 \def\VECTORSIZE(#1)#2{\expandafter\@VECTORSIZE(#1,,){#2}}
907 \def\@VECTORSIZE(#1,#2,#3,#4)#5{\ifx#3$\COPY{2}{#5}
908     \else\COPY{3}{#5}\fi\ignorespaces}

```

`\VECTORCOPY` `\VECTORCOPY(<#1,#2>)(<#3,#4>)` stores *#1* and *#2* in *#3* and *#4*.  
`\VECTORCOPY(<#1,#2,#3>)(<#4,#5#6>)` stores *#1*, *#2* and *#3* in *#4* and *#5* and *#6*.

```

909 \def\@@VECTORCOPY(#1,#2)(#3,#4){%
910     \COPY{#1}{#3}\COPY{#2}{#4}}
911
912 \def\@@@VECTORCOPY(#1,#2,#3)(#4,#5,#6){%
913     \COPY{#1}{#4}\COPY{#2}{#5}\COPY{#3}{#6}}
914
915 \def\VECTORCOPY(#1)(#2){%
916     \VECTORSIZE(#1){\cctr@size}
917     \ifnum\cctr@size=2
918         \@VECTORCOPY(#1)(#2)
919     \else \@@@VECTORCOPY(#1)(#2)\fi}

```

`\VECTORGLOBALCOPY` `\VECTORGLOBALCOPY` is the global version of `\VECTORCOPY`

```

920 \def\@@@VECTORGLOBALCOPY(#1,#2)(#3,#4){%
921     \GLOBALCOPY{#1}{#3}\GLOBALCOPY{#2}{#4}}
922
923 \def\@@@VECTORGLOBALCOPY(#1,#2,#3)(#4,#5,#6){%
924     \GLOBALCOPY{#1}{#4}\GLOBALCOPY{#2}{#5}\GLOBALCOPY{#3}{#6}}
925
926 \def\VECTORGLOBALCOPY(#1)(#2){%
927     \VECTORSIZE(#1){\cctr@size}
928     \ifnum\cctr@size=2
929         \@VECTORGLOBALCOPY(#1)(#2)
930     \else \@@@VECTORGLOBALCOPY(#1)(#2)\fi}

```

`\@OUTPUTVECTOR`

```

931 \def\@@@OUTPUTVECTOR(#1,#2){%

```



```

932 \VECTORGLOBALCOPY(#1,#2)(\ctr@outa,\ctr@outb)
933 \endgroup\VECTORCOPY(\ctr@outa,\ctr@outb)(#1,#2)}
934
935 \def\@@@OUTPUTVECTOR(#1,#2,#3){%
936 \VECTORGLOBALCOPY(#1,#2,#3)(\ctr@outa,\ctr@outb,\ctr@outc)
937 \endgroup\VECTORCOPY(\ctr@outa,\ctr@outb,\ctr@outc)(#1,#2,#3)}
938
939 \def\@OUTPUTVECTOR(#1){\VECTORSIZE(#1){\ctr@size}
940 \ifnum\ctr@size=2
941 \@@@OUTPUTVECTOR(#1)
942 \else \@@@OUTPUTVECTOR(#1)\fi}

```

\SCALARPRODUCT Scalar product of two vectors.

```

943 \def\@@SCALARPRODUCT(#1,#2)(#3,#4)#5{%
944 \MULTIPLY{#1}{#3}{#5}
945 \MULTIPLY{#2}{#4}\ctr@tempa
946 \ADD{#5}{\ctr@tempa}{#5}}
947
948 \def\@@@SCALARPRODUCT(#1,#2,#3)(#4,#5,#6)#7{%
949 \MULTIPLY{#1}{#4}{#7}
950 \MULTIPLY{#2}{#5}\ctr@tempa
951 \ADD{#7}{\ctr@tempa}{#7}
952 \MULTIPLY{#3}{#6}\ctr@tempa
953 \ADD{#7}{\ctr@tempa}{#7}}
954
955 \def\SCALARPRODUCT(#1)(#2)#3{%
956 \begingroup
957 \VECTORSIZE(#1){\ctr@size}
958 \ifnum\ctr@size=2
959 \@@SCALARPRODUCT(#1)(#2){#3}
960 \else \@@@SCALARPRODUCT(#1)(#2){#3}\fi\@OUTPUTSOL{#3}}

```

\DOTPRODUCT \DOTPRODUCT is an alias for \SCALARPRODUCT.

```

961 \let\DOTPRODUCT\SCALARPRODUCT

```

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

```

962 \def\@@VECTORPRODUCT(#1)(#2)(#3,#4){%
963 \let#3\undefined
964 \let#4\undefined
965 \ctr@Warncrossprod(#1)(#2)}
966
967 \def\@@@VECTORPRODUCT(#1,#2,#3)(#4,#5,#6)(#7,#8,#9){%
968 \DETERMINANT(#2,#3;#5,#6){#7}
969 \DETERMINANT(#3,#1;#6,#4){#8}
970 \DETERMINANT(#1,#2;#4,#5){#9}}
971
972 \def\VECTORPRODUCT(#1)(#2)(#3){%
973 \begingroup
974 \VECTORSIZE(#1){\ctr@size}

```

```

975     \ifnum\cctr@size=2
976         \@VECTORPRODUCT(#1)(#2)(#3)
977     \else \@@@VECTORPRODUCT(#1)(#2)(#3)\fi\@OUTPUTSOL{#3}}

```

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

```
978 \let\CROSSPRODUCT\VECTORPRODUCT
```

`\VECTORADD` Sum of two vectors.

```

979 \def\@@VECTORADD(#1,#2)(#3,#4)(#5,#6){%
980     \ADD{#1}{#3}{#5}
981     \ADD{#2}{#4}{#6}}
982
983 \def\@@@VECTORADD(#1,#2,#3)(#4,#5,#6)(#7,#8,#9){%
984     \ADD{#1}{#4}{#7}
985     \ADD{#2}{#5}{#8}
986     \ADD{#3}{#6}{#9}}
987
988 \def\VECTORADD(#1)(#2)(#3){%
989     \VECTORSIZE(#1){\cctr@size}
990     \ifnum\cctr@size=2
991         \@VECTORADD(#1)(#2)(#3)
992     \else \@@@VECTORADD(#1)(#2)(#3)\fi}

```

`\VECTORSUB` Difference of two vectors.

```

993 \def\@@VECTORSUB(#1,#2)(#3,#4)(#5,#6){%
994     \VECTORADD(#1,#2)(-#3,-#4)(#5,#6)}
995
996 \def\@@@VECTORSUB(#1,#2,#3)(#4,#5,#6)(#7,#8,#9){%
997     \VECTORADD(#1,#2,#3)(-#4,-#5,-#6)(#7,#8,#9)}
998
999 \def\VECTORSUB(#1)(#2)(#3){%
1000     \VECTORSIZE(#1){\cctr@size}
1001     \ifnum\cctr@size=2
1002         \@VECTORSUB(#1)(#2)(#3)
1003     \else \@@@VECTORSUB(#1)(#2)(#3)\fi}

```

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

```

1004 \def\@@VECTORABSVALUE(#1,#2)(#3,#4){%
1005     \ABSVALUE{#1}{#3}\ABSVALUE{#2}{#4}}
1006
1007 \def\@@@VECTORABSVALUE(#1,#2,#3)(#4,#5,#6){%
1008     \ABSVALUE{#1}{#4}\ABSVALUE{#2}{#5}\ABSVALUE{#3}{#6}}
1009
1010 \def\VECTORABSVALUE(#1)(#2){%
1011     \VECTORSIZE(#1){\cctr@size}
1012     \ifnum\cctr@size=2
1013         \@VECTORABSVALUE(#1)(#2)
1014     \else \@@@VECTORABSVALUE(#1)(#2)\fi}

```

`\SCALARVECTORPRODUCT` Scalar-vector product.

```

1015 \def\@@SCALARVECTORPRODUCT#1(#2,#3)(#4,#5){%
1016     \MULTIPLY{#1}{#2}{#4}
1017     \MULTIPLY{#1}{#3}{#5}}
1018
1019 \def\@@@SCALARVECTORPRODUCT#1(#2,#3,#4)(#5,#6,#7){%
1020     \MULTIPLY{#1}{#2}{#5}
1021     \MULTIPLY{#1}{#3}{#6}
1022     \MULTIPLY{#1}{#4}{#7}}
1023
1024 \def\SCALARVECTORPRODUCT#1(#2)(#3){%
1025     \VECTORSIZE(#2){\cctr@size}
1026     \ifnum\cctr@size=2
1027         \@@SCALARVECTORPRODUCT{#1}(#2)(#3)
1028     \else \@@@SCALARVECTORPRODUCT{#1}(#2)(#3)\fi}

```

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

```

1029 \def\VECTORNORM(#1)#2{%
1030     \begingroup
1031     \SCALARPRODUCT(#1)(#1){\cctr@temp}
1032     \SQAREROOT{\cctr@temp}{#2}\@OUTPUTSOL{#2}}

```

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

```

1033 \def\UNITVECTOR(#1)(#2){%
1034     \begingroup
1035     \VECTORNORM(#1){\cctr@tempa}
1036     \DIVIDE{1}{\cctr@tempa}{\cctr@tempa}
1037     \SCALARVECTORPRODUCT{\cctr@tempa}(#1)(#2)\@OUTPUTVECTOR{#2}}

```

**\TWOVECTORSANGLE** Angle between two vectors.

```

1038 \def\TWOVECTORSANGLE(#1)(#2)#3{%
1039     \begingroup
1040     \VECTORNORM(#1){\cctr@tempa}
1041     \VECTORNORM(#2){\cctr@tempb}
1042     \SCALARPRODUCT(#1)(#2){\cctr@tempc}
1043     \ifdim \cctr@tempa\p@ =\z@
1044         \let#3\undefined
1045         \cctr@Warnnoangle(#1)(#2)
1046     \else
1047         \ifdim \cctr@tempb\p@ =\z@
1048             \let#3\undefined
1049             \cctr@Warnnoangle(#1)(#2)
1050         \else
1051             \DIVIDE{\cctr@tempc}{\cctr@tempa}{\cctr@tempc}
1052             \DIVIDE{\cctr@tempc}{\cctr@tempb}{\cctr@tempc}
1053             \ARCCOS{\cctr@tempc}{#3}
1054         \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`.

```

1055 \def\@TDMATRIXCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1056     \COPY{#1}{\cctr@solAA}
1057     \COPY{#2}{\cctr@solAB}
1058     \COPY{#3}{\cctr@solAC}
1059     \COPY{#4}{\cctr@solBA}
1060     \COPY{#5}{\cctr@solBB}
1061     \COPY{#6}{\cctr@solBC}
1062     \COPY{#7}{\cctr@solCA}
1063     \COPY{#8}{\cctr@solCB}
1064     \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.

```

1065 \def\@TDMATRIXSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1066     \COPY{\cctr@solAA}{#1}
1067     \COPY{\cctr@solAB}{#2}
1068     \COPY{\cctr@solAC}{#3}
1069     \COPY{\cctr@solBA}{#4}
1070     \COPY{\cctr@solBB}{#5}
1071     \COPY{\cctr@solBC}{#6}
1072     \COPY{\cctr@solCA}{#7}
1073     \COPY{\cctr@solCB}{#8}
1074     \COPY{\cctr@solCC}{#9}}

```

`\@TDMATRIXGLOBALSOL`

```

1075 \def\@TDMATRIXGLOBALSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1076     \GLOBALCOPY{\cctr@solAA}{#1}
1077     \GLOBALCOPY{\cctr@solAB}{#2}
1078     \GLOBALCOPY{\cctr@solAC}{#3}
1079     \GLOBALCOPY{\cctr@solBA}{#4}
1080     \GLOBALCOPY{\cctr@solBB}{#5}
1081     \GLOBALCOPY{\cctr@solBC}{#6}
1082     \GLOBALCOPY{\cctr@solCA}{#7}
1083     \GLOBALCOPY{\cctr@solCB}{#8}
1084     \GLOBALCOPY{\cctr@solCC}{#9}}

```

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

```

1085 \def\@TDMATRIXNOSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1086     \let#1\undefined
1087     \let#2\undefined
1088     \let#3\undefined
1089     \let#4\undefined
1090     \let#5\undefined
1091     \let#6\undefined
1092     \let#7\undefined
1093     \let#8\undefined
1094     \let#9\undefined
1095     }

```

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

```
1096 \def\@TDMATRIXSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1097     \ifx\cctr@solAA\undefined
1098     \@TDMATRIXNOSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9)%
1099     \else
1100     \@TDMATRIXSOL(#1,#2,#3;#4,#5,#6;#7,#8,#9)\fi}
```

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

```
1101 \def\@NUMBERSOL#1{\COPY{\cctr@sol}{#1}}
```

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

```
1102 \def\MATRIXSIZE(#1)#2{\expandafter\MATRIXSIZE(#1;){#2}}
1103 \def\@MATRIXSIZE(#1;#2;#3;#4)#5{\ifx#3$\COPY{2}{#5}
1104     \else\COPY{3}{#5}\fi\ignorespaces}
```

`\MATRIXCOPY` Store a matrix in 4 or 9 commands.

```
1105 \def\@MATRIXCOPY(#1,#2;#3,#4)(#5,#6;#7,#8){%
1106     \COPY{#1}{#5}\COPY{#2}{#6}\COPY{#3}{#7}\COPY{#4}{#8}}
1107
1108 \def\@MATRIXCOPY(#1,#2;#3;#4,#5,#6;#7,#8,#9){%
1109     \@TDMATRIXCOPY(#1,#2;#3;#4,#5,#6;#7,#8,#9)
1110     \@TDMATRIXSOL}
1111
1112 \def\MATRIXCOPY(#1)(#2){%
1113     \MATRIXSIZE(#1){\cctr@size}
1114     \ifnum\cctr@size=2
1115     \@MATRIXCOPY(#1)(#2)
1116     \else \@MATRIXCOPY(#1)(#2)\fi}
```

`\MATRIXGLOBALCOPY` Global version of `\MATRIXCOPY`.

```
1117 \def\@MATRIXGLOBALCOPY(#1,#2;#3,#4)(#5,#6;#7,#8){%
1118     \GLOBALCOPY{#1}{#5}\GLOBALCOPY{#2}{#6}\GLOBALCOPY{#3}{#7}\GLOBALCOPY{#4}{#8}}
1119
1120 \def\@MATRIXGLOBALCOPY(#1,#2;#3;#4,#5,#6;#7,#8,#9){%
1121     \@TDMATRIXCOPY(#1,#2;#3;#4,#5,#6;#7,#8,#9)
1122     \@TDMATRIXGLOBALSOL}
1123
1124 \def\MATRIXGLOBALCOPY(#1)(#2){%
1125     \MATRIXSIZE(#1){\cctr@size}
1126     \ifnum\cctr@size=2
1127     \@MATRIXGLOBALCOPY(#1)(#2)
1128     \else \@MATRIXGLOBALCOPY(#1)(#2)\fi}
```

`\@OUTPUTMATRIX`

```
1129 \def\@OUTPUTMATRIX(#1,#2;#3,#4){%
1130     \MATRIXGLOBALCOPY(#1,#2;#3,#4)(\cctr@outa,\cctr@outb;\cctr@outc,\cctr@outd)
1131     \endgroup\MATRIXCOPY(\cctr@outa,\cctr@outb;\cctr@outc,\cctr@outd)(#1,#2;#3,#4)}
1132
1133 \def\@OUTPUTMATRIX(#1,#2;#3;#4,#5,#6;#7,#8,#9){%
```

```

1134 \MATRIXGLOBALCOPY(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1135 \ctr@outa,\ctr@outb,\ctr@outc;
1136 \ctr@outd,\ctr@oute,\ctr@outf;
1137 \ctr@outg,\ctr@outh,\ctr@outi)
1138 \endgroup\MATRIXCOPY(%
1139 \ctr@outa,\ctr@outb,\ctr@outc;
1140 \ctr@outd,\ctr@oute,\ctr@outf;
1141 \ctr@outg,\ctr@outh,\ctr@outi)(#1,#2,#3;#4,#5,#6;#7,#8,#9)}
1142
1143 \def\@OUTPUTMATRIX(#1){\MATRIXSIZE(#1){\ctr@size}
1144 \ifnum\ctr@size=2
1145 \@OUTPUTMATRIX(#1)
1146 \else \@@OUTPUTMATRIX(#1)\fi}

```

\TRANSPOSEMATRIX Matrix transposition.

```

1147 \def\@TRANSPOSEMATRIX(#1,#2,#3,#4)(#5,#6;#7,#8){%
1148 \COPY{#1}{#5}\COPY{#3}{#6}\COPY{#2}{#7}\COPY{#4}{#8}}
1149
1150 \def\@@TRANSPOSEMATRIX(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1151 \@TDMATRIXCOPY(#1,#4,#7;#2,#5,#8;#3,#6,#9)
1152 \@TDMATRIXSOL}
1153
1154 \def\TRANSPOSEMATRIX(#1)(#2){%
1155 \begingroup
1156 \MATRIXSIZE(#1){\ctr@size}
1157 \ifnum\ctr@size=2
1158 \@TRANSPOSEMATRIX(#1)(#2)
1159 \else \@@TRANSPOSEMATRIX(#1)(#2)\fi\@OUTPUTMATRIX(#2)}

```

\MATRIXADD Sum of two matrices.

```

1160 \def\@MATRIXADD(#1;#2)(#3;#4)(#5,#6;#7,#8){%
1161 \VECTORADD(#1)(#3)(#5,#6)
1162 \VECTORADD(#2)(#4)(#7,#8)}
1163
1164 \def\@@MATRIXADD(#1;#2;#3)(#4;#5;#6){%
1165 \VECTORADD(#1)(#4)(\ctr@solAA,\ctr@solAB,\ctr@solAC)
1166 \VECTORADD(#2)(#5)(\ctr@solBA,\ctr@solBB,\ctr@solBC)
1167 \VECTORADD(#3)(#6)(\ctr@solCA,\ctr@solCB,\ctr@solCC)
1168 \@TDMATRIXSOL}
1169
1170 \def\MATRIXADD(#1)(#2)(#3){%
1171 \begingroup
1172 \MATRIXSIZE(#1){\ctr@size}
1173 \ifnum\ctr@size=2
1174 \@MATRIXADD(#1)(#2)(#3)
1175 \else \@@MATRIXADD(#1)(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```

\MATRIXSUB Difference of two matrices.

```

1176 \def\@MATRIXSUB(#1;#2)(#3;#4)(#5,#6;#7,#8){%
1177 \VECTORSUB(#1)(#3)(#5,#6)

```

```

1178     \VECTORSUB(#2)(#4)(#7,#8)}
1179
1180 \def\@@@MATRIXSUB(#1;#2;#3)(#4;#5;#6){%
1181     \VECTORSUB(#1)(#4)(\cctr@solAA,\cctr@solAB,\cctr@solAC)
1182     \VECTORSUB(#2)(#5)(\cctr@solBA,\cctr@solBB,\cctr@solBC)
1183     \VECTORSUB(#3)(#6)(\cctr@solCA,\cctr@solCB,\cctr@solCC)
1184     \@TDMATRIXSOL}
1185
1186 \def\MATRIXSUB(#1)(#2)(#3){%
1187     \beginngroup
1188     \MATRIXSIZE(#1){\cctr@size}
1189     \ifnum\cctr@size=2
1190         \@@MATRIXSUB(#1)(#2)(#3)
1191     \else \@@@MATRIXSUB(#1)(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```

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

```

1192 \def\@@MATRIXABSVALUE(#1;#2)(#3;#4){%
1193     \VECTORABSVALUE(#1)(#3)\VECTORABSVALUE(#2)(#4)}
1194
1195 \def\@@@MATRIXABSVALUE(#1;#2;#3)(#4;#5;#6){%
1196     \VECTORABSVALUE(#1)(#4)\VECTORABSVALUE(#2)(#5)\VECTORABSVALUE(#3)(#6)}
1197
1198 \def\MATRIXABSVALUE(#1)(#2){%
1199     \beginngroup
1200     \MATRIXSIZE(#1){\cctr@size}
1201     \ifnum\cctr@size=2
1202         \@@MATRIXABSVALUE(#1)(#2)
1203     \else \@@@MATRIXABSVALUE(#1)(#2)\fi\@OUTPUTMATRIX(#2)}

```

**\MATRIXVECTORPRODUCT** Matrix-vector product.

```

1204 \def\@@MATRIXVECTORPRODUCT(#1;#2)(#3)(#4,#5){%
1205     \SCALARPRODUCT(#1)(#3){#4}
1206     \SCALARPRODUCT(#2)(#3){#5}}
1207
1208 \def\@@@MATRIXVECTORPRODUCT(#1;#2;#3)(#4)(#5,#6,#7){%
1209     \SCALARPRODUCT(#1)(#4){#5}
1210     \SCALARPRODUCT(#2)(#4){#6}
1211     \SCALARPRODUCT(#3)(#4){#7}}
1212
1213 \def\MATRIXVECTORPRODUCT(#1)(#2)(#3){%
1214     \beginngroup
1215     \MATRIXSIZE(#1){\cctr@size}
1216     \ifnum\cctr@size=2
1217         \@@MATRIXVECTORPRODUCT(#1)(#2)(#3)
1218     \else \@@@MATRIXVECTORPRODUCT(#1)(#2)(#3)\fi\@OUTPUTVECTOR(#3)}

```

**\VECTORMATRIXPRODUCT** Vector-matrix product.

```

1219 \def\@@VECTORMATRIXPRODUCT(#1)(#2,#3;#4,#5)(#6,#7){%
1220     \SCALARPRODUCT(#1)(#2,#4){#6}
1221     \SCALARPRODUCT(#1)(#3,#5){#7}}

```

```

1222
1223 \def\@@@VECTORMATRIXPRODUCT(#1,#2,#3)(#4,#5,#6)(#7){%
1224   \SCALARVECTORPRODUCT{#1}(#4)(#7)
1225   \SCALARVECTORPRODUCT{#2}(#5)(\cctr@tempa,\cctr@tempb,\cctr@tempc)
1226   \VECTORADD(#7)(\cctr@tempa,\cctr@tempb,\cctr@tempc)(#7)
1227   \SCALARVECTORPRODUCT{#3}(#6)(\cctr@tempa,\cctr@tempb,\cctr@tempc)
1228   \VECTORADD(#7)(\cctr@tempa,\cctr@tempb,\cctr@tempc)(#7)}
1229
1230 \def\VECTORMATRIXPRODUCT(#1)(#2)(#3){%
1231   \beginngroup
1232   \VECTORSIZE(#1){\cctr@size}
1233   \ifnum\cctr@size=2
1234     \@@VECTORMATRIXPRODUCT(#1)(#2)(#3)
1235   \else \@@@VECTORMATRIXPRODUCT(#1)(#2)(#3)\fi\@OUTPUTVECTOR(#3)}

```

\SCALARMATRIXPRODUCT Scalar-matrix product.

```

1236 \def\@@SCALARMATRIXPRODUCT#1(#2;#3)(#4,#5,#6,#7){%
1237   \SCALARVECTORPRODUCT{#1}(#2)(#4,#5)
1238   \SCALARVECTORPRODUCT{#1}(#3)(#6,#7)}
1239
1240 \def\@@@SCALARMATRIXPRODUCT#1(#2;#3;#4){%
1241   \SCALARVECTORPRODUCT{#1}(#2)(\cctr@solAA,\cctr@solAB,\cctr@solAC)
1242   \SCALARVECTORPRODUCT{#1}(#3)(\cctr@solBA,\cctr@solBB,\cctr@solBC)
1243   \SCALARVECTORPRODUCT{#1}(#4)(\cctr@solCA,\cctr@solCB,\cctr@solCC)
1244   \@TDMATRIXSOL}
1245
1246 \def\SCALARMATRIXPRODUCT#1(#2)(#3){%
1247   \beginngroup
1248   \MATRIXSIZE(#2){\cctr@size}
1249   \ifnum\cctr@size=2
1250     \@@SCALARMATRIXPRODUCT{#1}(#2)(#3)
1251   \else \@@@SCALARMATRIXPRODUCT{#1}(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```

\MATRIXPRODUCT Product of two matrices.

```

1252 \def\@@MATRIXPRODUCT(#1)(#2,#3;#4,#5)(#6,#7;#8,#9){%
1253   \MATRIXVECTORPRODUCT(#1)(#2,#4)(#6,#8)
1254   \MATRIXVECTORPRODUCT(#1)(#3,#5)(#7,#9)}
1255
1256 \def\@@@MATRIXPRODUCT(#1;#2;#3)(#4){%
1257   \VECTORMATRIXPRODUCT(#1)(#4)(\cctr@solAA,\cctr@solAB,\cctr@solAC)
1258   \VECTORMATRIXPRODUCT(#2)(#4)(\cctr@solBA,\cctr@solBB,\cctr@solBC)
1259   \VECTORMATRIXPRODUCT(#3)(#4)(\cctr@solCA,\cctr@solCB,\cctr@solCC)
1260   \@TDMATRIXSOL}
1261
1262 \def\MATRIXPRODUCT(#1)(#2)(#3){%
1263   \beginngroup
1264   \MATRIXSIZE(#1){\cctr@size}
1265   \ifnum\cctr@size=2
1266     \@@MATRIXPRODUCT(#1)(#2)(#3)
1267   \else \@@@MATRIXPRODUCT(#1)(#2)(#3)\fi\@OUTPUTMATRIX(#3)}

```



`\DETERMINANT` Determinant of a matrix.

```

1268 \def\@@DETERMINANT(#1,#2,#3,#4)#5{%
1269     \MULTIPLY{#1}{#4}{#5}
1270     \MULTIPLY{#2}{#3}{\ctr@tempa}
1271     \SUBTRACT{#5}{\ctr@tempa}{#5}}
1272
1273 \def\@@@DETERMINANT(#1,#2,#3,#4,#5,#6,#7,#8,#9){%
1274     \DETERMINANT(#5,#6,#8,#9){\ctr@det}\MULTIPLY{#1}{\ctr@det}{\ctr@sol}
1275     \DETERMINANT(#6,#4,#9,#7){\ctr@det}\MULTIPLY{#2}{\ctr@det}{\ctr@det}
1276     \ADD{\ctr@sol}{\ctr@det}{\ctr@sol}
1277     \DETERMINANT(#4,#5,#7,#8){\ctr@det}\MULTIPLY{#3}{\ctr@det}{\ctr@det}
1278     \ADD{\ctr@sol}{\ctr@det}{\ctr@sol}
1279     \@NUMBERSOL}
1280
1281 \def\DETERMINANT(#1)#2{%
1282     \begingroup
1283     \MATRIXSIZE(#1){\ctr@size}
1284     \ifnum\ctr@size=2
1285         \@@DETERMINANT(#1){#2}
1286     \else \@@@DETERMINANT(#1){#2}\fi\@OUTPUTSOL{#2}}

```

`\INVERSEMATRIX` Inverse of a matrix.

```

1287 \def\@@INVERSEMATRIX(#1,#2,#3,#4)(#5,#6,#7,#8){%
1288     \ifdim \ctr@det\p@ <\ctr@epsilon % Matrix is singular
1289         \let#5\undefined
1290         \let#6\undefined
1291         \let#7\undefined
1292         \let#8\undefined
1293         \ctr@Warningsingmatrix{#1}{#2}{#3}{#4}%
1294     \else \COPY{#1}{#8}
1295         \COPY{#4}{#5}
1296         \MULTIPLY{-1}{#3}{#7}
1297         \MULTIPLY{-1}{#2}{#6}
1298         \DIVIDE{1}{\ctr@det}{\ctr@det}
1299         \SCALARMATRIXPRODUCT{\ctr@det}{#5,#6,#7,#8)(#5,#6,#7,#8)
1300     \fi}
1301
1302 \def\@@@INVERSEMATRIX(#1,#2,#3,#4,#5,#6,#7,#8,#9){%
1303     \ifdim \ctr@det\p@ <\ctr@epsilon % Matrix is singular
1304         \@TDMATRIXNOSOL(\ctr@solAA,\ctr@solAB,\ctr@solAC;
1305             \ctr@solBA,\ctr@solBB,\ctr@solBC;
1306             \ctr@solCA,\ctr@solCB,\ctr@solCC)
1307         \ctr@WarningsingTdmatrix{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}{#9}%
1308     \else
1309         \@ADJMATRIX(#1,#2,#3,#4,#5,#6,#7,#8,#9)
1310         \@SCLRDIVVECT{\ctr@det}(\ctr@solAA,\ctr@solAB,\ctr@solAC)(%
1311             \ctr@solAA,\ctr@solAB,\ctr@solAC)
1312         \@SCLRDIVVECT{\ctr@det}(\ctr@solBA,\ctr@solBB,\ctr@solBC)(%
1313             \ctr@solBA,\ctr@solBB,\ctr@solBC)
1314         \@SCLRDIVVECT{\ctr@det}(\ctr@solCA,\ctr@solCB,\ctr@solCC)(%

```

```

1315                                     \cctr@solCA,\cctr@solCB,\cctr@solCC)
1316     \fi
1317     \@@TDMATRIXSOL}
1318
1319 \def\@SCLRDIVVECT#1(#2,#3,#4)(#5,#6,#7){%
1320     \DIVIDE{#2}{#1}{#5}\DIVIDE{#3}{#1}{#6}\DIVIDE{#4}{#1}{#7}}
1321
1322 \def\@ADJMATRIX(#1,#2,#3;#4,#5,#6;#7,#8,#9){%
1323     \DETERMINANT(#5,#6;#8,#9){\cctr@solAA}
1324     \DETERMINANT(#6,#4;#9,#7){\cctr@solBA}
1325     \DETERMINANT(#4,#5;#7,#8){\cctr@solCA}
1326     \DETERMINANT(#8,#9;#2,#3){\cctr@solAB}
1327     \DETERMINANT(#1,#3;#7,#9){\cctr@solBB}
1328     \DETERMINANT(#2,#1;#8,#7){\cctr@solCB}
1329     \DETERMINANT(#2,#3;#5,#6){\cctr@solAC}
1330     \DETERMINANT(#3,#1;#6,#4){\cctr@solBC}
1331     \DETERMINANT(#1,#2;#4,#5){\cctr@solCC}}
1332
1333 \def\@INVERSEMATRIX(#1)(#2){%
1334     \begingroup
1335     \DETERMINANT(#1){\cctr@det}
1336     \ABSVALUE{\cctr@det}{\cctr@@det}
1337     \MATRIXSIZE(#1){\cctr@size}
1338     \ifnum\cctr@size=2
1339         \@@INVERSEMATRIX(#1)(#2)
1340     \else
1341         \@@@INVERSEMATRIX(#1)(#2)\fi\@OUTPUTMATRIX(#2)}

```

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

```

1342 \def\@INCSYS#1#2{\cctr@WarnIncLinSys
1343     \let#1\undefined\let#2\undefined}
1344
1345 \def\@SOLPART#1#2#3#4{\cctr@WarnIndLinSys
1346     \DIVIDE{#1}{#2}{#3}
1347     \COPY{0}{#4}}
1348
1349 \def\@TDINCSYS(#1,#2,#3){\cctr@WarnIncTDLinSys
1350     \let#1\undefined
1351     \let#2\undefined
1352     \let#3\undefined}
1353
1354 \def\@@SOLVELINEARSYSTEM(#1,#2;#3,#4)(#5,#6)(#7,#8){%
1355     \DETERMINANT(#1,#2;#3,#4)\cctr@deta
1356     \DETERMINANT(#5,#2;#6,#4)\cctr@detb
1357     \DETERMINANT(#1,#5;#3,#6)\cctr@detc
1358     \ABSVALUE{\cctr@deta}{\cctr@@deta}
1359     \ABSVALUE{\cctr@detb}{\cctr@@detb}
1360     \ABSVALUE{\cctr@detc}{\cctr@@detc}
1361     \ifdim \cctr@@deta\p@>\cctr@epsilon% Regular matrix. Determinate system

```

```

1362         \DIVIDE{\cctr@detb}{\cctr@deta}{#7}
1363         \DIVIDE{\cctr@detc}{\cctr@deta}{#8}
1364     \else % Singular matrix    \cctr@deta=0
1365         \ifdim \cctr@detb\p@>\cctr@epsilon% Incompatible system
1366             \@INCSYS#7#8
1367     \else
1368         \ifdim \cctr@detc\p@>\cctr@epsilon% Incompatible system
1369             \@INCSYS#7#8
1370     \else
1371         \MATRIXABSVALUE(#1,#2,#3,#4)(\cctr@tempa,\cctr@tempb;
1372                                     \cctr@tempc,\cctr@tempd)
1373         \ifdim \cctr@tempa\p@ > \cctr@epsilon
1374             % Indeterminate system
1375             \@SOLPART{#5}{#1}{#7}{#8}
1376     \else
1377         \ifdim \cctr@tempb\p@ > \cctr@epsilon
1378             % Indeterminate system
1379             \@SOLPART{#5}{#2}{#8}{#7}
1380     \else
1381         \ifdim \cctr@tempc\p@ > \cctr@epsilon
1382             % Indeterminate system
1383             \@SOLPART{#6}{#3}{#7}{#8}
1384     \else
1385         \ifdim \cctr@tempd\p@ > \cctr@epsilon
1386             % Indeterminate system
1387             \@SOLPART{#6}{#4}{#8}{#7}
1388     \else
1389         \VECTORNORM(#5,#6){\cctr@tempa}
1390         \ifdim \cctr@tempa\p@ > \cctr@epsilon
1391             % Incompatible system
1392             \@INCSYS#7#8
1393     \else
1394         \cctr@WarnZeroLinSys
1395         \COPY{0}{#7}\COPY{0}{#8}
1396             % 0x=0 Indeterminate system
1397     \fi\fi\fi\fi\fi\fi\fi\fi}
1398
1399 \def\@@@SOLVELINEARSYSTEM(#1)(#2)(#3){%
1400     \DETERMINANT(#1){\cctr@det}
1401     \ABSVALUE{\cctr@det}{\cctr@det}
1402     \ifdim \cctr@det\p@<\cctr@epsilon
1403         \@TDINCSYS(#3)
1404     \else
1405         \@ADJMATRIX(#1)
1406         \MATRIXVECTORPRODUCT(\cctr@solAA,\cctr@solAB,\cctr@solAC;
1407                               \cctr@solBA,\cctr@solBB,\cctr@solBC;
1408                               \cctr@solCA,\cctr@solCB,\cctr@solCC)(#2)(#3)
1409         \@SCLRDIVVECT{\cctr@det}{#3}(#3)
1410     \fi}
1411

```

```

1412 \def\SOLVELINEARSYSTEM(#1)(#2)(#3){%
1413     \begingroup
1414     \MATRIXSIZE(#1){\cctr@size}
1415     \ifnum\cctr@size=2
1416         \@@SOLVELINEARSYSTEM(#1)(#2)(#3)
1417     \else
1418         \@@@SOLVELINEARSYSTEM(#1)(#2)(#3)
1419     \fi\@OUTPUTVECTOR(#3)}

```

## Predefined numbers

`\numberPI` The number  $\pi$

```
1420 \def\numberPI{3.14159}
```

`\numberTWOPI`  $2\pi$

```
1421 \MULTIPLY{\numberPI}{2}{\numberTWOPI}
```

`\numberHALFPI`  $\pi/2$

```
1422 \DIVIDE{\numberPI}{2}{\numberHALFPI}
```

`\numberTHREEHALFPI`  $3\pi/2$

```
1423 \MULTIPLY{\numberPI}{1.5}{\numberTHREEHALFPI}
```

`\numberTHIRDPI`  $\pi/3$

```
1424 \DIVIDE{\numberPI}{3}{\numberTHIRDPI}
```

`\numberQUARTERPI`  $\pi/4$

```
1425 \DIVIDE{\numberPI}{4}{\numberQUARTERPI}
```

`\numberFIFTHPI`  $\pi/5$

```
1426 \DIVIDE{\numberPI}{5}{\numberFIFTHPI}
```

`\numberSIXTHPI`  $\pi/6$

```
1427 \DIVIDE{\numberPI}{6}{\numberSIXTHPI}
```

`\numberE` The number  $e$

```
1428 \def\numberE{2.71828}
```

`\numberINVE`  $1/e$

```
1429 \DIVIDE{1}{\numberE}{\numberINVE}
```

`\numberETWO`  $e^2$

```
1430 \SQUARE{\numberE}{\numberETWO}
```

`\numberINVETWO`  $1/e^2$

```
1431 \SQUARE{\numberINVE}{\numberINVETWO}
```

`\numberLOGTEN`  $\log 10$

```
1432 \def\numberLOGTEN{2.30258}
```

```

\numberGOLD The golden ratio  $\phi$ 
1433 \def\numberGOLD{1.61803}

\numberINVGOLD  $1/\phi$ 
1434 \def\numberINVGOLD{0.61803}

\numberSQRTTWO  $\sqrt{2}$ 
1435 \def\numberSQRTTWO{1.41421}

\numberSQRTTHREE  $\sqrt{3}$ 
1436 \def\numberSQRTTHREE{1.73205}

\numberSQRTFIVE  $\sqrt{5}$ 
1437 \def\numberSQRTFIVE{2.23607}

\numberCOSXLV  $\cos 45^\circ$  (or  $\cos \pi/4$ )
1438 \def\numberCOSXLV{0.70711}

\numberCOSXXX  $\cos 30^\circ$  (or  $\cos \pi/6$ )
1439 \def\numberCOSXXX{0.86603}

1440 </calculator>

```

## 14 calculus

```

1441 (*calculus)
1442 \NeedsTeXFormat{LaTeX2e}
1443 \ProvidesPackage{calculus}[2014/02/20 v.2.0]

```

This package requires the calculator package.

```

1444 \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:

```

1445 \def\ccls@ErrorFuncDef#1{%
1446     \PackageError{calculus}%
1447         {\noexpand#1 command already defined}
1448         {The \noexpand#1 control sequence is already defined\MessageBreak
1449         If you want to redefine the \noexpand#1 command as a
1450         function\MessageBreak
1451         please, use the \noexpand\renewfunction command}}

```

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

```

1452 \def\ccls@ErrorFuncUnDef#1{%
1453     \PackageError{calculus}%

```

```

1454     {\noexpand#1 command undefined}
1455     {The \noexpand#1 control sequence is not currently defined\MessageBreak
1456       If you want to define the \noexpand#1 command as a function\MessageBreak
1457       please, use the \noexpand\newfunction command}}

```

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

```

1458 \def\ccls@InfoFuncEns#1{%
1459   \PackageInfo{calculus}%
1460   {\noexpand#1 command already defined\MessageBreak
1461     the \noexpand\ensurefunction command will not redefine it}}

```

### For polar functions

```

1462 \def\ccls@ErrorPFuncDef#1{%
1463   \PackageError{calculus}%
1464     {\noexpand#1 command already defined}
1465     {The \noexpand#1 control sequence is already defined\MessageBreak
1466       If you want to redefine the \noexpand#1
1467       command as a polar function\MessageBreak
1468       please, use the \noexpand\renewpolarfunction command}}
1469
1470 \def\ccls@ErrorPFuncUnDef#1{%
1471   \PackageError{calculus}%
1472     {\noexpand#1 command undefined}
1473     {The \noexpand#1 control sequence
1474       is not currently defined.\MessageBreak
1475       If you want to define the \noexpand#1 command as a polar
1476       function\MessageBreak
1477       please, use the \noexpand\newpolarfunction command}}
1478
1479 \def\ccls@InfoPFuncEns#1{%
1480   \PackageInfo{calculus}%
1481   {\noexpand#1 command already defined\MessageBreak
1482     the \noexpand\ensurepolarfunction command does not redefine it}}

```

### For vector functions

```

1483 \def\ccls@ErrorVFuncDef#1{%
1484   \PackageError{calculus}%
1485     {\noexpand#1 command already defined}
1486     {The \noexpand#1 control sequence is already defined\MessageBreak
1487       If you want to redefine the \noexpand#1 command as a vector
1488       function\MessageBreak
1489       please, use the \noexpand\renewvectorfunction command}}
1490
1491 \def\ccls@ErrorVFuncUnDef#1{%
1492   \PackageError{calculus}%
1493     {\noexpand#1 command undefined}
1494     {The \noexpand#1 control sequence is not currently
1495       defined.\MessageBreak
1496       If you want to define the \noexpand#1 command as a vector
1497       function\MessageBreak

```

```

1498         please, use the \noexpand\newvectorfunction command}}
1499
1500 \def\ccls@InfoVFuncEns#1{%
1501     \PackageInfo{calculus}%
1502     {\noexpand#1 command already defined\MessageBreak
1503     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  $y$  and his derivative  $Dy$  from the  $t$  variable.

```

1504 \def\newfunction#1#2{%
1505     \ifx #1\undefined
1506         \ccls@deffunction{#1}{#2}
1507     \else
1508         \ccls@ErrorFuncDef{#1}
1509     \fi}

```

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

```

1510 \def\renewfunction#1#2{%
1511     \ifx #1\undefined
1512         \ccls@ErrorFuncUnDef{#1}
1513     \else
1514         \ccls@deffunction{#1}{#2}
1515     \fi}

```

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

```

1516 \def\ensurefunction#1#2{%
1517     \ifx #1\undefined\ccls@deffunction{#1}{#2}
1518     \else
1519         \ccls@InfoFuncEns{#1}
1520     \fi}

```

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

```

1521 \def\forcefunction#1#2{%
1522     \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.

```

1523 \def\ccls@deffunction#1#2{%
1524     \def#1##1##2##3{%
1525         \begingroup
1526         \def\t{##1}%
1527         #2
1528         \xdef##2{\y}%
1529         \xdef##3{\Dy}%
1530     \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.

```
1531 \def\newpolarfunction#1#2{%
1532     \ifx #1\undefined
1533         \ccls@defpolarfunction{#1}{#2}
1534     \else
1535         \ccls@ErrorPFuncDef{#1}
1536     \fi}
```

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

```
1537 \def\renewpolarfunction#1#2{%
1538     \ifx #1\undefined
1539         \ccls@ErrorPFuncUnDef{#1}
1540     \else
1541         \ccls@defpolarfunction{#1}{#2}
1542     \fi}
```

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

```
1543 \def\ensurepolarfunction#1#2{%
1544     \ifx #1\undefined\ccls@defpolarfunction{#1}{#2}
1545     \else
1546         \ccls@InfoPFuncEns{#1}
1547     \fi}
```

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

```
1548 \def\forcepolarfunction#1#2{%
1549     \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.

```
1550 \def\ccls@defpolarfunction#1#2{%
1551     \def##1##2##3##4##5{%
1552         \begingroup
1553             \def\t{##1}
1554             #2
1555             \COS{\t}\ccls@cost
1556             \MULTIPLY\r\ccls@cost{\x}
1557             \SIN{\t}\ccls@sint
1558             \MULTIPLY\r\ccls@sint{\y}
1559             \MULTIPLY\ccls@cost\Dr\Dx
1560             \SUBTRACT{\Dx}{\y}{\Dx}
1561             \MULTIPLY\ccls@sint\Dr\Dy
1562             \ADD{\Dy}{\x}{\Dy}
1563             \xdef##2{\x}
1564             \xdef##3{\Dx}
1565             \xdef##4{\y}
1566             \xdef##5{\Dy}
1567             \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.

```
1568 \def\newvectorfunction#1#2{%
1569     \ifx #1\undefined
1570         \ccls@defvectorfunction{#1}{#2}
1571     \else
1572         \ccls@ErrorVFuncDef{#1}
1573     \fi}
```

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

```
1574 \def\renewvectorfunction#1#2{%
1575     \ifx #1\undefined
1576         \ccls@ErrorVFuncUnDef{#1}
1577     \else
1578         \ccls@defvectorfunction{#1}{#2}
1579     \fi}
```

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

```
1580 \def\ensurevectorfunction#1#2{%
1581     \ifx #1\undefined\ccls@defvectorfunction{#1}{#2}
1582     \else
1583         \ccls@InfoVFuncEns{#1}
1584     \fi}
```

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

```
1585 \def\forcevectorfunction#1#2{%
1586     \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.

```
1587 \def\ccls@defvectorfunction#1#2{%
1588     \def##1##2##3##4##5{%
1589         \begingroup
1590         \def\t{##1}
1591         #2
1592         \xdef##2{\x}
1593         \xdef##3{Dx}
1594         \xdef##4{y}
1595         \xdef##5{Dy}
1596         \endgroup}\ignorespaces}
```

## 14.3 Polynomials

### Linear (first degree) polynomials

`\newlpoly` The `\newlpoly{#1}{#2}{#3}` instruction defines the linear polynomial

$$\#1 = \#2 + \#3t.$$

```

1597 \def\newlpoly#1#2#3{%
1598   \newfunction{#1}{%
1599     \ccls@lpoly{#2}{#3}}}
```

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

```

1600 \def\renewlpoly#1#2#3{%
1601   \renewfunction{#1}{%
1602     \ccls@lpoly{#2}{#3}}}
```

`\ensurelpoly`

```

1603 \def\ensurelpoly#1#2#3{%
1604   \ensurefunction{#1}{%
1605     \ccls@lpoly{#2}{#3}}}
```

`\forcelpoly`

```

1606 \def\forcelpoly#1#2#3{%
1607   \forcefunction{#1}{%
1608     \ccls@lpoly{#2}{#3}}}
```

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

```

1609 \def\ccls@lpoly#1#2{%
1610   \MULTIPLY{#2}{\t}{\y}
1611   \ADD{\y}{#1}{\y}
1612   \COPY{#2}{\Dy}}
```

### Quadratic polynomials

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

```

1613 \def\newqpoly#1#2#3#4{%
1614   \newfunction{#1}{%
1615     \ccls@qpoly{#2}{#3}{#4}}}
```

`\renewqpoly`

```

1616 \def\renewqpoly#1#2#3#4{%
1617   \renewfunction{#1}{%
1618     \ccls@qpoly{#2}{#3}{#4}}}
```

`\ensureqpoly`

```

1619 \def\ensureqpoly#1#2#3#4{%
1620   \ensurefunction{#1}{%
1621     \ccls@qpoly{#2}{#3}{#4}}}
```

`\forceqpoly`

```

1622 \def\forceqpoly#1#2#3#4{%
1623   \forcefunction{#1}{%
1624     \ccls@qpoly{#2}{#3}{#4}}}
```

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

```

1625 \def\ccls@qpoly#1#2#3{%
1626     \MULTIPLY{\t}{#3}{\y}
1627     \MULTIPLY{2}{\y}{\Dy}
1628     \ADD{#2}{\Dy}{\Dy}
1629     \ADD{#2}{\y}{\y}
1630     \MULTIPLY{\t}{\y}{\y}
1631     \ADD{#1}{\y}{\y}}

```

### Cubic polynomials

`\newcpoly` The `\newcpoly{#1}{#2}{#3}{#4}{#5}` instruction defines the cubic polynomial

$$\#1 = \#2 + \#3t + \#4t^2 + \#5t^3.$$

```

1632 \def\newcpoly#1#2#3#4#5{%
1633     \newfunction{#1}{%
1634         \ccls@cpoly{#2}{#3}{#4}{#5}}

```

`\renewcpoly`

```

1635 \def\renewcpoly#1#2#3#4#5{%
1636     \renewfunction{#1}{%
1637         \ccls@cpoly{#2}{#3}{#4}{#5}}

```

`\ensurecpoly`

```

1638 \def\ensurecpoly#1#2#3#4#5{%
1639     \ensurefunction{#1}{%
1640         \ccls@cpoly{#2}{#3}{#4}{#5}}

```

`\forcecpoly`

```

1641 \def\forcecpoly#1#2#3#4#5{%
1642     \forcefunction{#1}{%
1643         \ccls@cpoly{#2}{#3}{#4}{#5}}

```

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

```

1644 \def\ccls@cpoly#1#2#3#4{%
1645     \MULTIPLY{\t}{#4}{\y}
1646     \MULTIPLY{3}{\y}{\Dy}
1647     \ADD{#3}{\y}{\y}
1648     \MULTIPLY{2}{#3}{\ccls@temp}
1649     \ADD{\ccls@temp}{\Dy}{\Dy}
1650     \MULTIPLY{\t}{\y}{\y}
1651     \MULTIPLY{\t}{\Dy}{\Dy}
1652     \ADD{#2}{\y}{\y}
1653     \ADD{#2}{\Dy}{\Dy}
1654     \MULTIPLY{\t}{\y}{\y}
1655     \ADD{#1}{\y}{\y}
1656     }

```

## 14.4 Elementary functions

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

```
1657 \newfunction{\ONEfunction}{%
1658     \COPY{1}{\y}
1659     \COPY{0}{\Dy}}
```

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

```
1660 \newfunction{\ZEROfunction}{%
1661     \COPY{0}{\y}
1662     \COPY{0}{\Dy}}
```

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

```
1663 \newfunction{\IDENTITYfunction}{%
1664     \COPY{t}{\y}
1665     \COPY{1}{\Dy}}
```

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

```
1666 \newfunction{\RECIPROCALfunction}{%
1667     \DIVIDE{1}{t}{\y}
1668     \SQUARE{\y}{\Dy}
1669     \MULTIPLY{-1}{\Dy}{\Dy}}
```

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

```
1670 \newfunction{\SQUAREfunction}{%
1671     \SQUARE{t}{\y}
1672     \MULTIPLY{2}{t}{\Dy}}
```

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

```
1673 \newfunction{\CUBEfunction}{%
1674     \SQUARE{t}{\Dy}
1675     \MULTIPLY{t}{\Dy}{\y}
1676     \MULTIPLY{3}{\Dy}{\Dy}}
```

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

```
1677 \newfunction{\SQRTfunction}{%
1678     \SQRT{t}{\y}
1679     \DIVIDE{0.5}{\y}{\Dy}}
```

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

```
1680 \newfunction{\EXPfunction}{%
1681     \EXP{t}{\y}
1682     \COPY{\y}{\Dy}}
```

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

```
1683 \newfunction{\COSfunction}{%
1684     \COS{t}{\y}
1685     \SIN{t}{\Dy}
1686     \MULTIPLY{-1}{\Dy}{\Dy}}
```

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

1687 `\newfunction{\SINfunction}{%`

1688 `\SIN{t}{y}`

1689 `\COS{t}{Dy}}`

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

1690 `\newfunction{\TANfunction}{%`

1691 `\TAN{t}{y}`

1692 `\COS{t}{Dy}`

1693 `\SQUARE{Dy}{Dy}`

1694 `\DIVIDE{1}{Dy}{Dy}}`

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

1695 `\newfunction{\COTfunction}{%`

1696 `\COTAN{t}{y}`

1697 `\SIN{t}{Dy}`

1698 `\SQUARE{Dy}{Dy}`

1699 `\DIVIDE{-1}{Dy}{Dy}}`

`\COSHfunction` The `\COSHfunction`:  $y(t) = \cosh t$ ,  $y'(t) = \sinh t$

1700 `\newfunction{\COSHfunction}{%`

1701 `\COSH{t}{y}`

1702 `\SINH{t}{Dy}}`

`\SINHfunction` The `\SINHfunction`:  $y(t) = \sinh t$ ,  $y'(t) = \cosh t$

1703 `\newfunction{\SINHfunction}{%`

1704 `\SINH{t}{y}`

1705 `\COSH{t}{Dy}}`

`\TANHfunction` The `\TANHfunction`:  $y(t) = \tanh t$ ,  $y'(t) = 1/(\cosh t)^2$

1706 `\newfunction{\TANHfunction}{%`

1707 `\TANH{t}{y}`

1708 `\COSH{t}{Dy}`

1709 `\SQUARE{Dy}{Dy}`

1710 `\DIVIDE{1}{Dy}{Dy}}`

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

1711 `\newfunction{\COTHfunction}{%`

1712 `\COTANH{t}{y}`

1713 `\SINH{t}{Dy}`

1714 `\SQUARE{Dy}{Dy}`

1715 `\DIVIDE{-1}{Dy}{Dy}}`

`\LOGfunction` The `\LOGfunction`:  $y(t) = \log t$ ,  $y'(t) = 1/t$

1716 `\newfunction{\LOGfunction}{%`

1717 `\LOG{t}{y}`

1718 `\DIVIDE{1}{t}{Dy}}`

`\HEAVISIDEfunction` The `\HEAVISIDEfunction`:  $y(t) = \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{if } t \geq 0 \end{cases}, y'(t) = 0$

```

1719 \newfunction{\HEAVISIDEfunction}{%
1720     \ifdim \t\p@<\z@ \COPY{0}{\y}\else\COPY{1}{\y}\fi
1721     \COPY{0}{\Dy}}

```

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

```

1722 \newfunction{\ARCSINfunction}{%
1723     \ARCSIN{\t}{\y}
1724     \SQUARE{\t}{\yy}
1725     \SUBTRACT{1}{\yy}{\yy}
1726     \SQRT{\yy}{\Dy}
1727     \DIVIDE{1}{\Dy}{\Dy}}

```

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

```

1728 \newfunction{\ARCCOSfunction}{%
1729     \ARCCOS{\t}{\y}
1730     \SQUARE{\t}{\yy}
1731     \SUBTRACT{1}{\yy}{\yy}
1732     \SQRT{\yy}{\Dy}
1733     \DIVIDE{-1}{\Dy}{\Dy}}

```

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

```

1734 \newfunction{\ARCTANfunction}{%
1735     \ARCTAN{\t}{\y}
1736     \SQUARE{\t}{\yy}
1737     \ADD{1}{\yy}{\yy}
1738     \DIVIDE{1}{\yy}{\Dy}}

```

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

```

1739 \newfunction{\ARCCOTfunction}{%
1740     \ARCCOT{\t}{\y}
1741     \SQUARE{\t}{\yy}
1742     \ADD{1}{\yy}{\yy}
1743     \DIVIDE{-1}{\yy}{\Dy}}

```

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

```

1744 \newfunction{\ARSINHfunction}{%
1745     \ARSINH{\t}{\y}
1746     \SQUARE{\t}{\yy}
1747     \ADD{1}{\yy}{\yy}
1748     \SQRT{\yy}{\Dy}
1749     \DIVIDE{1}{\Dy}{\Dy}}

```

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

```

1750 \newfunction{\ARCOSHfunction}{%
1751     \ARCOSH{\t}{\y}

```

```

1752 \SQUARE{\t}{\yy}
1753 \SUBTRACT{\yy}{1}{\yy}
1754 \SQRT{\yy}{\Dy}
1755 \DIVIDE{1}{\Dy}{\Dy}

```

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

```

1756 \newfunction{\ARTANHfunction}{%
1757 \ARTANH{\t}{\y}
1758 \SQUARE{\t}{\yy}
1759 \SUBTRACT{1}{\yy}{\yy}
1760 \DIVIDE{1}{\yy}{\Dy}}

```

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

```

1761 \newfunction{\ARCOTHfunction}{%
1762 \ARCOTH{\t}{\y}
1763 \SQUARE{\t}{\yy}
1764 \SUBTRACT{1}{\yy}{\yy}
1765 \DIVIDE{1}{\yy}{\Dy}}

```

## 14.5 Operations with functions

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

```

1766 \def\CONSTANTfunction#1#2{%
1767 \def##1##2##3{%
1768 \xdef##2{#1}%
1769 \xdef##3{0}}}

```

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

```

1770 \def\SUMfunction#1#2#3{%
1771 \def##1##2##3{%
1772 \begingroup
1773 #1{##1}{\ccls@SUMf}{\ccls@SUMDf}%
1774 #2{##1}{\ccls@SUMg}{\ccls@SUMDg}%
1775 \ADD{\ccls@SUMf}{\ccls@SUMg}{\ccls@SUMfg}
1776 \ADD{\ccls@SUMDf}{\ccls@SUMDg}{\ccls@SUMDfg}
1777 \xdef##2{\ccls@SUMfg}%
1778 \xdef##3{\ccls@SUMDfg}%
1779 \endgroup}\ignorespaces}

```

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

```

1780 \def\SUBTRACTfunction#1#2#3{%
1781 \def##1##2##3{%
1782 \begingroup
1783 #1{##1}{\ccls@SUBf}{\ccls@SUBDf}%
1784 #2{##1}{\ccls@SUBg}{\ccls@SUBDg}%
1785 \SUBTRACT{\ccls@SUBf}{\ccls@SUBg}{\ccls@SUBfg}
1786 \SUBTRACT{\ccls@SUBDf}{\ccls@SUBDg}{\ccls@SUBDfg}
1787 \xdef##2{\ccls@SUBfg}%
1788 \xdef##3{\ccls@SUBDfg}%
1789 \endgroup}\ignorespaces}

```

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

```

1790 \def\PRODUCTfunction#1#2#3{%
1791     \def#3##1##2##3{%
1792         \begingroup
1793             #1{##1}{\ccls@PROf}{\ccls@PRODf}%
1794             #2{##1}{\ccls@PROg}{\ccls@PRODg}%
1795             \MULTIPLY{\ccls@PROf}{\ccls@PROg}{\ccls@PROfg}
1796             \MULTIPLY{\ccls@PROf}{\ccls@PRODg}{\ccls@PROfdg}
1797             \MULTIPLY{\ccls@PRODf}{\ccls@PROg}{\ccls@PROdfg}
1798             \ADD{\ccls@PROfdg}{\ccls@PROdfg}{\ccls@PRODfg}
1799             \xdef##2{\ccls@PROfg}%
1800             \xdef##3{\ccls@PRODfg}%
1801         \endgroup}\ignorespaces}

```

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

```

1802 \def\QUOTIENTfunction#1#2#3{%
1803     \def#3##1##2##3{%
1804         \begingroup
1805             #1{##1}{\ccls@QUOf}{\ccls@QUODf}%
1806             #2{##1}{\ccls@QUOg}{\ccls@QUODg}%
1807             \DIVIDE{\ccls@QUOf}{\ccls@QUOg}{\ccls@QUOfg}
1808             \MULTIPLY{\ccls@QUOf}{\ccls@QUODg}{\ccls@QUOfdg}
1809             \MULTIPLY{\ccls@QUODf}{\ccls@QUOg}{\ccls@QUODfg}
1810             \SUBTRACT{\ccls@QUODfg}{\ccls@QUOfdg}{\ccls@QUOnum}
1811             \SQUARE{\ccls@QUOg}{\ccls@qsquaretempg}
1812             \DIVIDE{\ccls@QUOnum}{\ccls@qsquaretempg}{\ccls@QUODfg}
1813             \xdef##2{\ccls@QUOfg}%
1814             \xdef##3{\ccls@QUODfg}%
1815         \endgroup}\ignorespaces}

```

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

```

1816 \def\COMPOSITIONfunction#1#2#3{% #3=#1(#2)
1817     \def#3##1##2##3{%
1818         \begingroup
1819             #2{##1}{\ccls@COMg}{\ccls@COMDg}%
1820             #1{\ccls@COMg}{\ccls@COMf}{\ccls@COMDf}%
1821             \MULTIPLY{\ccls@COMDg}{\ccls@COMDf}{\ccls@COMDf}
1822             \xdef##2{\ccls@COMf}%
1823             \xdef##3{\ccls@COMDf}%
1824         \endgroup}\ignorespaces}

```

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

```

1825 \def\SCALEfunction#1#2#3{%
1826     \def#3##1##2##3{%
1827         \begingroup
1828             #2{##1}{\ccls@SCFf}{\ccls@SCFDf}%
1829             \MULTIPLY{#1}{\ccls@SCFf}{\ccls@SCFaf}
1830             \MULTIPLY{#1}{\ccls@SCFDf}{\ccls@SCFDaf}
1831             \xdef##2{\ccls@SCFaf}%
1832             \xdef##3{\ccls@SCFDaf}%

```



1833                   \endgroup}\ignorespaces}

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

```
1834 \def\SCALEVARIABLEfunction#1#2#3{%
1835       \def#3##1##2##3{%
1836       \begingroup%
1837           \MULTIPLY{#1}{##1}{\ccls@SCVat}
1838           #2{\ccls@SCVat}{\ccls@SCVf}{\ccls@SCVDf}%
1839           \MULTIPLY{#1}{\ccls@SCVDf}{\ccls@SCVDf}
1840           \xdef##2{\ccls@SCVf}%
1841           \xdef##3{\ccls@SCVDf}%
1842       \endgroup}\ignorespaces}
```

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

```
1843 \def\POWERfunction#1#2#3{%
1844       \def#3##1##2##3{%
1845       \begingroup
1846           #1{##1}{\ccls@POWf}{\ccls@POWdf}%
1847           \POWER{\ccls@POWf}{#2}{\ccls@POWfn}
1848           \SUBTRACT{#2}{1}{\ccls@nminusone}
1849           \POWER{\ccls@POWf}{\ccls@nminusone}{\ccls@POWdfn}
1850           \MULTIPLY{#2}{\ccls@POWdfn}{\ccls@POWdfn}
1851           \MULTIPLY{\ccls@POWdfn}{\ccls@POWdf}{\ccls@POWdfn}
1852           \xdef##2{\ccls@POWfn}%
1853           \xdef##3{\ccls@POWdfn}%
1854       \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.

```
1855 \def\LINEARCOMBINATIONfunction#1#2#3#4#5{%
1856       \def#5##1##2##3{%
1857       \begingroup
1858           #2{##1}{\ccls@LINF}{\ccls@LINDf}%
1859           #4{##1}{\ccls@LING}{\ccls@LINDg}%
1860           \MULTIPLY{#1}{\ccls@LINF}{\ccls@LINF}
1861           \MULTIPLY{#3}{\ccls@LING}{\ccls@LING}
1862           \MULTIPLY{#1}{\ccls@LINDf}{\ccls@LINDf}
1863           \MULTIPLY{#3}{\ccls@LINDg}{\ccls@LINDg}
1864           \ADD{\ccls@LINF}{\ccls@LING}{\ccls@LINafbg}
1865           \ADD{\ccls@LINDf}{\ccls@LINDg}{\ccls@LINDafbg}
1866           \xdef##2{\ccls@LINafbg}%
1867           \xdef##3{\ccls@LINDafbg}%
1868       \endgroup}\ignorespaces}
```

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

```
1869 \def\POLARfunction#1#2{%
1870       \PRODUCTfunction{#1}{\COSfunction}{\ccls@polarx}
1871       \PRODUCTfunction{#1}{\SINfunction}{\ccls@polary}
1872       \PARAMETRICfunction{\ccls@polarx}{\ccls@polary}{#2}}
```

```

\PARAMETRICfunction \PARAMETRICfunction defines the parametric curve #3. #1 and #2 are the components func-
tions (two previously defined functions).
1873 \def\PARAMETRICfunction#1#2#3{%
1874     \def#3##1##2##3##4##5{%
1875         #1{##1}{##2}{##3}
1876         #2{##1}{##4}{##5}}

\VECTORfunction \VECTORfunction: an alias of \PARAMETRICfunction.
1877 \let\VECTORfunction\PARAMETRICfunction

1878 % </calculus>

```

## Change History

v1.0	General: First public version . . . . . 1	New commands: \ARCSIN, \ARCCOS, \ARCTAN, \ARCCOT . . . . . 51
v1.0a	General: calculator.dtx modified to make it autoinstallable. calculus.dtx embedded in calculus.dtx . . . . . 1	New commands: \ARSINHfunction, \ARCOSHfunction, \ARTANHfunction, \ARCOTHfunction . . . . . 78
v2.0	General: new calculator.dtx and calcula- tor.ins files . . . . . 1	New commands: \ARSINH, \ARCOSH, \ARTANH, \ARCOTH . . . . . 54
	New commands: \ARCSINfunction, \ARCCOSfunction, \ARCTANfunction, \ARCCOTfunction . . . . . 78	New commands: \DOTPRODUCT, \VECTORPRODUCT, \CROSSPRODUCT . . . 57
		New commands: \LENGTHADD, \LENGTHSUBTRACT . . . . . 35
		Trivial error in documentation corrected 69

## 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		
\# . . . . . 701	\@@@OUTPUTMATRIX 1133, 1146	\@@@VECTORADD . . . . . 983, 992
\@@@DETERMINANT . 1273, 1286	\@@@OUTPUTVECTOR . . 935, 942	\@@@VECTORCOPY . . . . . 912, 919
\@@@INVERSEMATRIX 1302, 1341	\@@@SCALARMATRIXPRODUCT . . . . . 1240, 1251	\@@@VECTORGLOBALCOPY 923, 930
\@@@MATRIXABSVALUE 1195, 1203	\@@@SCALARPRODUCT . 948, 960	\@@@VECTORMATRIXPRODUCT . . . . . 1223, 1235
\@@@MATRIXADD . . . 1164, 1175	\@@@SCALARVECTORPRODUCT . . . . . 1019, 1028	\@@@VECTORPRODUCT . 967, 977
\@@@MATRIXCOPY . . 1108, 1116	\@@@SOLVELINEARSYSTEM . . . . . . . 1399, 1418	\@@@VECTORSUB . . . . . 996, 1003
\@@@MATRIXGLOBALCOPY . . . . . . . . 1120, 1128	\@@@TRANPOSEMATRIX . . . . .	\@@@DEGREESCOS . . . . . 500, <u>565</u>
\@@@MATRIXPRODUCT 1256, 1267	. . . . . 1150, 1159	\@@@DEGREESCOT . . . . . 502, <u>569</u>
\@@@MATRIXSUB . . . 1180, 1191	\@@@TRUNCATE . . 278, 280, 283	\@@@DEGREESSIN . . . . . 499, <u>563</u>
\@@@MATRIXVECTORPRODUCT . . . . . 1208, 1218	\@@@VECTORABSVLUE 1007, 1014	\@@@DEGREESTAN . . . . . 501, <u>567</u>
		\@@@DETERMINANT . . 1268, 1285
		\@@@EXP . . . . . 573, <u>574</u>



\ccls@defvectorfunction	\ccls@QUOg	\cctr@expy
..... 1570,	..... 1806, 1807, 1809, 1811	596, 597, 604, 605
1578, 1581, 1586, <u>1587</u>	\ccls@QUOnum	\cctr@lengtha
\ccls@ErrorFuncDef 1445, 1508	.... 1810, 1812	.....
\ccls@ErrorFuncUnDef ...	\ccls@SCFaf	... 4, 145–148, 150,
..... 1452, 1512	.... 1829, 1831	151, 182, 188, 189,
\ccls@ErrorPFuncDef	\ccls@SCFDaf	191, 193, 194, 199, 201
....	.... 1830, 1832	\cctr@lengthb
..... 1462, 1535	\ccls@SCFDf	.....
\ccls@ErrorPFuncUnDef ..	.... 1828, 1830	5, 149, 150, 183, 184,
..... 1470, 1539	\ccls@SCFf	188, 192, 193, 200,
\ccls@ErrorVFuncDef	.... 1837, 1838	202, 359, 360, 367,
....	\ccls@SCVdf	369, 704, 705, 712,
..... 1483, 1572	.. 1838, 1839, 1841	714, 756, 758, 769, 771
\ccls@ErrorVFuncUnDef ..	\ccls@SCVf	\cctr@log
..... 1491, 1576	.... 1838, 1840	..... 580–582
\ccls@InfoFuncEns 1458, 1519	\ccls@sint	\cctr@loga
\ccls@InfoPFuncEns 1479, 1546	.... 1557, 1558, 1561	..... 697, 699
\ccls@InfoVFuncEns 1500, 1583	\ccls@SUBDf	\cctr@logmaxnum
\ccls@LINAfbg ... 1864, 1866	.... 1783, 1786	.. 8, 102,
\ccls@LINDafbg .. 1865, 1867	\ccls@SUBDfg	587, 632, 645, 658, 669
\ccls@LINDf . 1858, 1862, 1865	.... 1786, 1788	\cctr@logx
\ccls@LINDg . 1859, 1863, 1865	\ccls@SUBDg	..... 698, 699
\ccls@LINF . 1858, 1860, 1864	.... 1784, 1786	\cctr@minust
\ccls@LING . 1859, 1861, 1864	\ccls@SUBf	637, 638, 650, 651
\ccls@lpoly	.... 1783, 1785	\cctr@endec
..... 1599,	\ccls@SUBfg	... 164, 165, 174
1602, 1605, 1608, <u>1609</u>	.... 1785, 1787	\cctr@outa
\ccls@nminusone . 1848, 1849	\ccls@SUBg	132, 133, 135,
\ccls@polarx	.... 1784, 1785	932, 933, 936, 937,
.... 1870, 1872	\ccls@SUMDf	1130, 1131, 1135, 1139
\ccls@polarx	.... 1773, 1776	\cctr@outb
.... 1871, 1872	\ccls@SUMDfg	.... 134, 135,
\ccls@POWdf	.... 1776, 1778	932, 933, 936, 937,
..... 1846, 1851	\ccls@SUMDg	1130, 1131, 1135, 1139
\ccls@POWdfn 1849–1851, 1853	.... 1774, 1776	\cctr@outc
\ccls@POWf . 1846, 1847, 1849	\ccls@SUMf	.... 936, 937,
\ccls@POWfn	.... 1773, 1775	1130, 1131, 1135, 1139
..... 1847, 1852	\ccls@SUMfg	\cctr@outd
\ccls@PRODf	.... 1775, 1777	.....
.... 1793, 1797	\ccls@SUMg	.. 1130, 1131, 1136, 1140
\ccls@PRODfg 1797, 1798, 1800	.... 1774, 1775	\cctr@oute
\ccls@PRODg	\ccls@temp	.... 1136, 1140
.... 1794, 1796	.... 1648, 1649	\cctr@outf
\ccls@PROf . 1793, 1795, 1796	\cctr@@det	.... 1137, 1141
\ccls@PROfdg	.... 1288,	\cctr@outh
.... 1796, 1798	1303, 1336, 1401, 1402	.... 1137, 1141
\ccls@PROfg	\cctr@@deta	\cctr@Q
.... 1795, 1799	.... 1358, 1361	..... 163, 173, 175
\ccls@PROg . 1794, 1795, 1797	\cctr@@detb	\cctr@sign
\ccls@qpoly	.... 1359, 1365	... 157–160, 175
..... 1615,	\cctr@@detc	\cctr@size
1618, 1621, 1624, <u>1625</u>	.... 1360, 1368	916, 917, 927,
\ccls@squaretempg 1811, 1812	\cctr@absval	928, 939, 940, 957,
\ccls@QUODf	..... 586,	958, 974, 975, 989,
..... 1805, 1809	587, 631, 632, 644,	990, 1000, 1001, 1011,
\ccls@QUODfg	645, 657, 658, 668, 669	1012, 1025, 1026,
.....	\cctr@ae	1113, 1114, 1125,
.. 1809, 1810, 1812, 1814	.. 685, 686, 690, 691	1126, 1143, 1144,
\ccls@QUODg	\cctr@det	1156, 1157, 1172,
.... 1806, 1808	... 1274–1278,	1173, 1188, 1189,
\ccls@QUOf . 1805, 1807, 1808	1298, 1299, 1310,	1200, 1201, 1215,
\ccls@QUOfdg	1312, 1314, 1335,	1216, 1232, 1233,
.... 1808, 1810	1336, 1400, 1401, 1409	1248, 1249, 1264,
\ccls@QUOfg	\cctr@deta	1265, 1283, 1284,
..... 1807, 1813	.....	1337, 1338, 1414, 1415
	.. 1355, 1358, 1362–1364	
	\cctr@detb	
	.. 1356, 1359, 1362	
	\cctr@detc	
	.. 1357, 1360, 1363	
	\cctr@epsilon	
	6, 360, 576,	
	680, 705, 758, 1288,	
	1303, 1361, 1365,	
	1368, 1373, 1377,	
	1381, 1385, 1390, 1402	
	\cctr@expminux	
	.....	
	.... 638, 639, 651, 652	
	\cctr@expt	
	595, 596, 603, 604	
	\cctr@expx	
	636, 639, 649, 652	

\cctr@sol .....	397, 398, 400, 402,	\cctr@tempr .....	162, 166, 167, 169, 172
. 1101, 1274, 1276, 1278	403, 405, 408, 439,	\cctr@tempw ....	361–363,
\cctr@solAA .....	440, 442, 446, 447,	366, 367, 703, 706,	707, 709–711, 716,
.. 1056, 1066, 1076,	449, 450, 452, 453,	759, 761, 763, 765, 767	
1097, 1165, 1181,	455, 456, 458, 459,	\cctr@tempx 737–739, 745–747	
1241, 1257, 1304,	610, 611, 613, 616,	\cctr@tempxw .....	707–709
1310, 1311, 1323, 1406	619, 622, 814, 815,	\cctr@tempxx 739, 740, 747, 748	
\cctr@solAB .. 1057, 1067,	817, 820, 823, 826,	\cctr@tempy 753, 754, 757,	
1077, 1165, 1181,	829, 832, 835, 846–	760, 763, 764, 767–770	
1241, 1257, 1304,	848, 858–860, 874–	\cctr@tempz .....	362–
1310, 1311, 1326, 1406	876, 891–893, 902,	365, 761, 762, 764–766	
\cctr@solAC .. 1058, 1068,	903, 945, 946, 950–	\cctr@Warnbigarccos ....	
1078, 1165, 1181,	953, 1035–1037, 1040,	..... 36, 786, 790	
1241, 1257, 1304,	1043, 1051, 1225–	\cctr@Warnbigarcsin ....	
1310, 1311, 1329, 1406	1228, 1270, 1271,	..... 30, 730, 734	
\cctr@solBA .. 1059, 1069,	1371, 1373, 1389, 1390	\cctr@Warnbigartanh ....	
1079, 1166, 1182,	\cctr@tempB 877, 879, 894, 896	..... 48, 868, 872	
1242, 1258, 1305,	\cctr@tempb .....	\cctr@Warncrossprod 119, 965	
1312, 1313, 1324, 1407	. 202, 203, 218, 220,	\cctr@Warndivzero .. 9, 186	
\cctr@solBB .. 1060, 1070,	301, 302, 305, 306,	\cctr@WarnIncLinSys 71, 1342	
1080, 1166, 1182,	314–317, 319, 323,	\cctr@WarnIncTDLinSys ..	
1242, 1258, 1305,	328, 329, 382, 383,	..... 74, 1349	
1312, 1313, 1327, 1407	389, 390, 392, 393,	\cctr@WarnIndLinSys 78, 1345	
\cctr@solBC .. 1061, 1071,	428, 429, 434, 435,	\cctr@Warninfcotan .....	92, 470, 558
1081, 1166, 1182,	512, 513, 516, 517,	\cctr@Warninfexp ... 98,	
1242, 1258, 1305,	520, 521, 523, 524,	588, 633, 646, 659, 670	
1312, 1313, 1330, 1407	541, 544, 545, 547,	\cctr@Warninfexpb . 106, 577	
\cctr@solCA .. 1062, 1072,	548, 798, 799, 803,	\cctr@Warninflog .. 113, 681	
1082, 1167, 1183,	804, 848–850, 860–	\cctr@Warninfktan .....	86, 420, 424, 533, 537
1243, 1259, 1306,	862, 875, 877, 892,	\cctr@Warnnoangle .....	125, 1045, 1049
1314, 1315, 1325, 1408	894, 1041, 1047, 1052,	\cctr@Warnnogcd .... 14, 320	
\cctr@solCB .. 1063, 1073,	1225–1228, 1371, 1377	\cctr@Warnnointexp .. 24, 211	
1083, 1167, 1183,	\cctr@tempC 878, 879, 895, 896	\cctr@Warnnopostrad .. 18, 356	
1243, 1259, 1306,	\cctr@tempc .....	\cctr@Warnsingmatrix 61, 1293	
1314, 1315, 1328, 1408	. 219, 220, 300, 302,	\cctr@WarnsingTMatrix .	
\cctr@solCC .. 1064, 1074,	303, 315, 317, 328,	..... 66, 1307	
1084, 1167, 1183,	330, 415, 416, 528,	\cctr@Warnsmallarcosh 42, 856	
1243, 1259, 1306,	529, 876, 878, 893,	\cctr@Warnsmallarcoth 54, 888	
1314, 1315, 1331, 1408	895, 1042, 1051–1053,	\cctr@WarnZeroLinSys 82, 1394	
\cctr@tanhdn .....	1225–1228, 1372, 1381	\COMPOSITIONfunction 25, 1816	
.... 663, 664, 673, 675	\cctr@tempD .....	\CONSTANTfunction . 24, 1766	
\cctr@tanhnum .....	. 155, 162, 167, 169, 172	\COPY 5, 130, 132, 135, 138,	
.... 662, 664, 674, 675	\cctr@tempd 156, 162, 170,	141, 144, 157–160,	
\cctr@temp 243, 246, 252–	172, 230, 239, 1372, 1385	163, 164, 169, 173,	
254, 296, 297, 1031, 1032	\cctr@tempdif .....	189, 222, 223, 247,	
\cctr@tempA .....	711, 712		
267, 268	\cctr@tempe 299, 300, 306, 308		
\cctr@tempa .....	\cctr@tempexp .....		
201,	209, 210		
203, 222, 224, 226,	\cctr@tempoldw .... 706, 711		
276, 277, 313, 315,	\cctr@tempoldy .... 760, 770		
316, 318, 326–330,	\cctr@tempq 162, 163, 172, 173		

248, 256, 262, 263,  
 285–289, 297, 317,  
 323, 327, 329, 330,  
 334, 342, 352, 358,  
 361, 365, 375, 377,  
 379, 398, 400, 440,  
 443, 466, 478, 487,  
 489, 498, 505, 507,  
 509, 554, 706, 716,  
 720, 723, 726, 752,  
 757, 759, 760, 766,  
 776, 779, 782, 812,  
 874, 891, 904, 907,  
 908, 910, 913, 1056–  
 1064, 1066–1074,  
 1101, 1103, 1104,  
 1106, 1148, 1294,  
 1295, 1347, 1395,  
 1612, 1658, 1659,  
 1661, 1662, 1664,  
 1665, 1682, 1720, 1721  
 \COS . . . 11, 413, 464, 763,  
 1555, 1684, 1689, 1692  
 \COSfunction . 23, 1683, 1870  
 \COSH . . . . . 13, 629, 663,  
 674, 1701, 1705, 1708  
 \COSHfunction . . . . . 23, 1700  
 \COT . . . . . 12, 462  
 \COTAN . . . . . 1696  
 \COTANH . . . . . 1712  
 \COTfunction . . . . . 23, 1695  
 \COTH . . . . . 13, 666  
 \COTHfunction . . . . . 23, 1711  
 \CROSSPRODUCT . . . . . 17, 978  
 \CUBE . . . . . 7, 206  
 \CUBEfunction . . . . . 23, 1673

**D**

\DEGREESCOS 12, 500, 552, 566  
 \DEGREESCOT . . . . . 12, 502, 570  
 \DEGREESSIN . . . . . 12,  
 499, 513, 521, 524, 529  
 \DEGREESTAN . . . . . 12,  
501, 541, 548, 556, 568  
 \DEGtoRAD . 13, 475, 516, 544  
 \DETERMINANT . . . 21, 968–  
 970, 1268, 1323–1331,  
 1335, 1355–1357, 1400  
 \DIVIDE . . . . . 6, 153, 203,  
 218, 305, 336, 345,

346, 362, 364, 403,  
 406, 409, 445, 446,  
 448, 449, 451, 452,  
 454, 455, 457, 458,  
 460, 473, 475, 561,  
 571, 595, 603, 611,  
 613, 614, 616, 617,  
 619, 620, 622, 623,  
 626, 640, 653, 664,  
 675, 685, 699, 708,  
 738, 746, 764, 798,  
 803, 817, 820, 823,  
 826, 829, 832, 835,  
 837, 880, 897, 1036,  
 1051, 1052, 1298,  
 1320, 1346, 1362,  
 1363, 1422, 1424–  
 1427, 1429, 1667,  
 1679, 1694, 1699,  
 1710, 1715, 1718,  
 1727, 1733, 1738,  
 1743, 1749, 1755,  
 1760, 1765, 1807, 1812  
 \DOTPRODUCT . . . . . 17, 961  
 \Dr . . . . . 1559, 1561  
 \Dx . . . . . 1559, 1560, 1564, 1593  
 \Dy . . . . . 1529, 1561, 1562,  
 1566, 1595, 1612,  
 1627, 1628, 1646,  
 1649, 1651, 1653,  
 1659, 1662, 1665,  
 1668, 1669, 1672,  
 1674–1676, 1679,  
 1682, 1685, 1686,  
 1689, 1692–1694,  
 1697–1699, 1702,  
 1705, 1708–1710,  
 1713–1715, 1718,  
 1721, 1726, 1727,  
 1732, 1733, 1738,  
 1743, 1748, 1749,  
 1754, 1755, 1760, 1765

**E**

\ensurecpoly . . . . . 27, 1638  
 \ensurefunction 29, 1461,  
1516, 1604, 1620, 1639  
 \ensurelpoly . . . . . 27, 1603  
 \ensurepolarfunction . . .  
 . . . . . 30, 1482, 1543

\ensureqpoly . . . . . 27, 1619  
 \ensurevectorfunction . .  
 . . . . . 29, 1503, 1580  
 \EXP . . . . . 11, 573, 636,  
 638, 649, 651, 707, 1681  
 \EXPfunction . . . . . 23, 1680

**F**

\FLOOR . . . . . 8, 261  
 \forcecpoly . . . . . 27, 1641  
 \forcefunction . . . . . 29,  
1521, 1607, 1623, 1642  
 \forcelpoly . . . . . 27, 1606  
 \forcepolarfunction 30, 1548  
 \forcepoly . . . . . 27, 1622  
 \forcevectorfunction 29, 1585  
 \FRACTIONALPART . . . . . 8, 262  
 \FRACTIONSIMPLIFY . 10, 340

**G**

\GCD . . . . . 10, 311, 333, 344  
 \GLOBALCOPY . . . . .  
 . . . . . 131, 132–134, 921,  
 924, 1076–1084, 1118

**H**

\HEAVISIDEfunction . 23, 1719

**I**

\IDENTITYfunction . 23, 1663  
 \ifcase . . . . . 284  
 \INTEGERDIVISION . . . . .  
 . . . . . 9, 228, 243, 246  
 \INTEGERPART . . . . . 7,  
 209, 247, 261, 267, 276  
 \INTEGERQUOTIENT . . . 9, 244  
 \INVERSEMATRIX . . . . 21, 1287

**L**

\LCM . . . . . 10, 332  
 \LENGTHADD . . . . . 15, 191, 196  
 \LENGTHDIVIDE . . . . . 15, 197  
 \LENGTHSUBTRACT . . . . 15, 195  
 \LINEARCOMBINATIONfunction  
 . . . . . 26, 1855  
 \LOG . . . . . 11, 580,  
 677, 691, 850, 862,  
 877, 878, 894, 895, 1717  
 \LOGfunction . . . . . 23, 1716

**M**

\MATRIXABSVALUE 22, 1192, 1371



<code>\SINfunction</code> . 23, <u>1687</u> , 1871	1557, 1590, 1610,	<code>\VECTORMATRIXPRODUCT</code> ...
<code>\SINH</code> ..... 13, <u>642</u> , 662,	1626, 1630, 1645,	..... <u>1219</u> , 1257–1259
673, 1702, 1704, 1713	1650, 1651, 1654,	<code>\VECTORNORM</code> ... 17, <u>1029</u> ,
<code>\SINHfunction</code> ..... 23, <u>1703</u>	1664, 1667, 1671,	1035, 1040, 1041, 1389
<code>\SOLVELINEARSYSTEM</code> . 22, <u>1342</u>	1672, 1674, 1675,	<code>\VECTORPRODUCT</code> . 17, <u>962</u> , 978
<code>\SQRT</code> ..... 10, <u>372</u> , 739,	1678, 1681, 1684,	<code>\VECTORSIZE</code> 906, 916, 927,
747, 848, 860, 1678,	1685, 1688, 1689,	939, 957, 974, 989,
1726, 1732, 1748, 1754	1691, 1692, 1696,	1000, 1011, 1025, 1232
<code>\SQRTfunction</code> ..... 23, <u>1677</u>	1697, 1701, 1702,	<code>\VECTORSUB</code> .... 16, <u>993</u> ,
<code>\SQUARE</code> ..... 7, <u>205</u> , 402,	1704, 1705, 1707,	1177, 1178, 1181–1183
597, 605, 610, 814,	1708, 1712, 1713,	
846, 858, 1430, 1431,	1717, 1718, 1720,	<b>X</b>
1668, 1671, 1674,	1723, 1724, 1729,	<code>\x</code> ... 1556, 1562, 1563, 1592
1693, 1698, 1709,	1730, 1735, 1736,	
1714, 1724, 1730,	1740, 1741, 1745,	<b>Y</b>
1736, 1741, 1746,	1746, 1751, 1752,	<code>\y</code> .... 1528, 1558, 1560,
1752, 1758, 1763, 1811	1757, 1758, 1762, 1763	1565, 1594, 1610,
<code>\SQUAREfunction</code> ... 23, <u>1670</u>	<code>\TAN</code> ..... 11, <u>417</u> , 468, 1691	1611, 1626, 1627,
<code>\SQUAREROOT</code> 10, <u>349</u> , 372, 1032	<code>\TANfunction</code> ..... 23, <u>1690</u>	1629–1631, 1645–
<code>\SUBTRACT</code> ..... 6, <u>152</u> ,	<code>\TANH</code> ..... 13, <u>655</u> , 1707	1647, 1650, 1652,
180, 237, 255, 268,	<code>\TANHfunction</code> ..... 23, <u>1706</u>	1654, 1655, 1658,
300, 366, 389, 392,	<code>\textit</code> ..... 701	1661, 1664, 1667,
404, 407, 410, 434,	<code>\TRANSPOSEMATRIX</code> .. 19, <u>1147</u>	1668, 1671, 1675,
447, 450, 453, 456,	<code>\TRUNCATE</code> ..... 8, <u>272</u>	1678, 1679, 1681,
459, 484, 495, 520,	<code>\TWOVECTORSANGLE</code> .. 18, <u>1038</u>	1682, 1684, 1688,
523, 547, 625, 652,		1691, 1696, 1701,
692, 703, 710, 711,	<b>U</b>	1704, 1707, 1712,
737, 742, 750, 762,	<code>\UNITVECTOR</code> ..... 17, <u>1033</u>	1717, 1720, 1723,
765, 767, 793, 800,		1729, 1735, 1740,
805, 842, 859, 876,	<b>V</b>	1745, 1751, 1757, 1762
879, 893, 896, 1271,	<code>\VECTORABSVALUE</code> .....	<code>\yy</code> ... 1724–1726, 1730–
1560, 1725, 1731,	... 18, <u>1004</u> , 1193, 1196	1732, 1736–1738,
1753, 1759, 1764,	<code>\VECTORADD</code> .... 16, <u>979</u> ,	1741–1743, 1746–
1785, 1786, 1810, 1848	994, 997, 1161, 1162,	1748, 1752–1754,
<code>\SUBTRACTfunction</code> . 25, <u>1780</u>	1165–1167, 1226, 1228	1758–1760, 1763–1765
<code>\SUMfunction</code> ..... 25, <u>1770</u>	<code>\VECTORCOPY</code> 16, <u>909</u> , 933, 937	
	<code>\VECTORfunction</code> ... 28, <u>1877</u>	<b>Z</b>
<b>T</b>	<code>\VECTORGLOBALCOPY</code> .....	<code>\ZEROfunction</code> ..... 23, <u>1660</u>
<code>\t</code> .... 1526, 1553, 1555,	..... <u>920</u> , 932, 936	