Laplace

P!\K

COLLABORATORS

	TITLE : Laplace	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	P!\K	April 18, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Laplace

1.1 libraries

```
                - Laplace Manual ---------------------------------------------- ↩
                    Libraries -

9) Libraries

   Laplace offers the ability to process external files. These files are plain
ASCII files and may be edited  with every texteditor. They will be  processed,
just as if you  entered the contents  in a window. The  results won't be  dis-
played, so this is only useful for constant and function definitions.  Library
files have a trailing .lh and  are located in the Data/Include directory.  Use
the include() command to gain access to a library.
> include("linalg.lh");

   If you want a library to be available on startup, add a line  include("lib-
name.lh"); to the  file Data/Include/init.lh. This  way you can  load all  re-
quired libraries with the single command include("init.lh");.


                linalg.lh
                :

                 *
                 addcols()
                 *
                 addrows()
                 *
                 angle()

                 *
                 multcol()
                 *
                 multrow()
                 *
                 swapcols()

                 *
                 swaprows()
                 *
```

```
 umatrix()


analysis.lh
:

 *
 randomcpoly()
 *
 randompoly()
 *
 randomqpoly()

 *
 randomrpoly()


statistics.lh
:

 *
 average()
 *
 error()
 *
 sigma()


physic.lh
:


math.lh
:


---------------------------------------------------------------- © by P!\K -
```

1.2 liblinalg

```
- Laplace Manual ------------------------------------------------ ↩
    linalg.lh -

9.1) linalg.lh


 *
 addcols()
 *
 addrows()
 *
 angle()

 *
 multcol()
```

```
                *
                multrow()
                *
                swapcols()

                *
                swaprows()
                *
                umatrix()


------------------------------------------------------------------ © by P!\K -
```

1.3 fkt-addcols

```
- Laplace Manual ------------------------------------------------ addcols() -

addcols()

sypnosis:
   addcols(M , c1 , c2 , x )

arguments:
   M : matrix
   c1, c2 : integer between 1  and cols  M
   x : number

result:
   matrix

   Returns the matrix M , except that column c1  is replaced by

                          (column c1) + x * (column c2)


------------------------------------------------------------------ © by P!\K -
```

1.4 fkt-addrows

```
- Laplace Manual ------------------------------------------------ addrows() -

addrows()

sypnosis:
   addrows(M , r1 , r2 , x )

arguments:
   M : matrix
   r1, r2 : integer between 1  and rows  M
   x : number

result:
```

```
   matrix

   Returns the matrix M , except that row r1  is replaced by

                            (row r1) + x * (row r2)

------------------------------------------------------------------ © by P!\K -
```

1.5 fkt-angle

```
- Laplace Manual ------------------------------------------------- angle() -

angle()

sypnosis:
   angle(a , b )

arguments:
   a,b : vector

result:
   number

   Calculate the angle, enclosed by the vectors a  and b :

                  angle  x = arccos acot b / abs(a)*abs(b)

------------------------------------------------------------------ © by P!\K -
```

1.6 fkt-multcol

```
- Laplace Manual ------------------------------------------------ multcol() -

multcol()

sypnosis:
   multcol(M , c , x )

arguments:
   M : matrix
   c : integer between 1  and cols  M
   x : number

result:
   matrix

   Returns the matrix M , except that column c  is multiplied by x .

------------------------------------------------------------------ © by P!\K -
```

1.7 fkt-multrow

```
- Laplace Manual ------------------------------------------------ multrow() -

multrow()

sypnosis:
   multrow(M , r , x )

arguments:
   M : matrix
   r : integer between 1  and rows  M
   x : number

result:
   matrix

   Returns the matrix M , except that rows r  is multiplied by x .


------------------------------------------------------------------- © by P!\K -
```

1.8 fkt-swapcols

```
- Laplace Manual ------------------------------------------------ swapcols() -

swapcols()

sypnosis:
   swapcols(M , c1 , c2 )

arguments:
   M : matrix
   c1, c2 : integer between 1  and cols  M

result:
   matrix

   Return the matrix M , except that columns c1  and c2  are swapped.


------------------------------------------------------------------- © by P!\K -
```

1.9 fkt-swaprows

```
- Laplace Manual ------------------------------------------------ swaprows() -

swaprows()

sypnosis:
   swaprows(M , r1 , r2 )
```

```
arguments:
   M : matrix
   r1, r2 : integer between 1  and rows  M

result:
   matrix

   Return the matrix M , except that rows r1  and r2  are swapped.


------------------------------------------------------------------- © by P!\K -
```

1.10 fkt-umatrix

```
- Laplace Manual ------------------------------------------------ umatrix() -

umatrix()

sypnosis:
   umatrix(n )

arguments:
   n : positive integer

result:
   matrix

   This will create a standart matrix of the given size.


------------------------------------------------------------------- © by P!\K -
```

1.11 libanalysis

```
                - Laplace Manual ---------------------------------------------  ↩
                   analysis.lh -

9.2) analysis.lh


                 *
                 randomcpoly()
                 *
                 randompoly()
                 *
                 randomqpoly()

                 *
                 randomrpoly()


------------------------------------------------------------------- © by P!\K -
```

1.12 fkt-randomcpoly

```
- Laplace Manual ------------------------------------------- randomcpoly() -

randomcpoly()

sypnosis:
   randomcpoly(n )

arguments:
   n : positive integer

result:
   number

   This creates a polynom in  x  of n -th  degree with random complex coeffi-
cients a_i , with -1?re a_i?1  and -1?im a_i?1 .
> f(x) = randomcpoly(2)
>    => 0.5-0.3i+(0.3+0.5i)*x-(0.7-0.4i)*x^2


------------------------------------------------------------------ © by P!\K -
```

1.13 fkt-randompoly

```
- Laplace Manual ------------------------------------------- randompoly() -

randompoly()

sypnosis:
   randompoly(n )

arguments:
   n : positive integer

result:
   number

   This creates a polynom in  x  of n -th  degree with random integer coeffi-
cients a_i , with -10?a_i?10 .
> f(x) = randompoly(2)
>    => 5+3*x-7*x^2


------------------------------------------------------------------ © by P!\K -
```

1.14 fkt-randomqpoly

```
- Laplace Manual ------------------------------------------- randomqpoly() -

randomqpoly()
```

```
sypnosis:
   randomqpoly(n )

arguments:
   n : positive integer

result:
   number

   This creates a polynom in x  of  n -th degree with random rational coeffi-
cients a_i / b_i , with -10?a_i?10  and 1?b_i?10 .
> f(x) = randomqpoly(2)
>    => 3/7+5/8*x-1/3*x^2


------------------------------------------------------------------- © by P!\K -
```

1.15 fkt-randomrpoly

```
- Laplace Manual ------------------------------------------- randomrpoly() -

randomrpoly()

sypnosis:
   randomrpoly(n )

arguments:
   n : positive integer

result:
   number

   This creates a polynom in x  of n -th degree with random real coefficients
a_i , with -1?a_i?1 .
> f(x) = randomrpoly(2)
>    => 0.5+0.3*x-0.7*x^2


------------------------------------------------------------------- © by P!\K -
```

1.16 libstatistics

```
                - Laplace Manual ------------------------------------------  ↩
                   statistics.lh -

9.3) statistics.lh


                *
                average()
                *
                error()
```

```
                *
                sigma()

------------------------------------------------------------------ © by P!\K -
```

1.17 fkt-average

```
- Laplace Manual ------------------------------------------------ average() -

average()

sypnosis:
   average(A )

arguments:
   A : array of numbers

result:
   number

   Returns the arithmetic average of the array's contents.

            average  A = 1 / n *sum  x_i, A = {x_1, x_2,...x_n}

------------------------------------------------------------------ © by P!\K -
```

1.18 fkt-error

```
- Laplace Manual ------------------------------------------------- error() -

error()

sypnosis:
   error(A )

arguments:
   A : array of numbers

result:
   number

   Calculate the statistical error of the array's contents, which is

                  error  A = sigma  A / sqrt(count  A)

------------------------------------------------------------------ © by P!\K -
```

1.19 fkt-sigma

```
- Laplace Manual ---------------------------------------------- sigma() -

sigma()

sypnosis:
   sigma(A )

arguments:
   A : array of numbers

result:
   number

   Calculate the standard variation of the array's contents, which is  defined
as

   sigma  A = sqrt(sum ((x - y)^2) / (count  A - 1)), with y = average  A


------------------------------------------------------------------ © by P!\K -
```

1.20 libphysic

```
- Laplace Manual ---------------------------------------------- physic.lh -

9.4) physic.lh

   This library defines some useful physical constants, like G , h  etc. There
are currently no functions available.



------------------------------------------------------------------ © by P!\K -
```

1.21 libmath

```
- Laplace Manual ------------------------------------------------ math.lh -

9.5) math.lh

   There are currently no functions available.



------------------------------------------------------------------ © by P!\K -
```