# KANT/KASH tutorial

## http://www.math.tu-berlin.de/~kant/

## *Fourth RISC/SCIEnce training school – 2009*

LESSENI SYLLA

TU Berlin - Fakultät II

Institut für Mathematik Straße des

17. Juni 136 D-10623 Berlin, Germany

lesseni@math.tu-berlin.de

# Lectures outline

# Lectures outline

- Overview of the KANT/KASH system

# Lectures outline

- Overview of the KANT/KASH system
- Applications

# Lectures outline

- Overview of the KANT/KASH system
- Applications
- KANT/KASH SCSCP Package

# Overview of the KANT system

- What is KANT/KASH?

# Overview of the KANT system

- What is KANT/KASH?
- Starting and Leaving KASH3

# Overview of the KANT system

- What is KANT/KASH?
- Starting and Leaving KASH3
- KASH3 - Features

# Overview of the KANT system

- What is KANT/KASH?
- Starting and Leaving KASH3
- KASH3 - Features
- KASH3 - Type System

# Overview of the KANT system

- What is KANT/KASH?
- Starting and Leaving KASH3
- KASH3 - Features
- KASH3 - Type System
- KASH3 - Help System

# Overview of the KANT system

- What is KANT/KASH?
- Starting and Leaving KASH3
- KASH3 - Features
- KASH3 - Type System
- KASH3 - Help System
- Operations

# Overview of the KANT system

- What is KANT/KASH?
- Starting and Leaving KASH3
- KASH3 - Features
- KASH3 - Type System
- KASH3 - Help System
- Operations
- Polynomials

# Overview of the KANT system

- What is KANT/KASH?
- Starting and Leaving KASH3
- KASH3 - Features
- KASH3 - Type System
- KASH3 - Help System
- Operations
- Polynomials
- Kant Database

# What is KANT/KASH?

# What is KANT/KASH?

- KANT: stands for *Computational Algebraic Number Theory*, with a slight hint at its German origin (Immanuel Kant).
  A sophisticated computer algebra system for computations in algebraic number fields, algebraic function fields and local fields.
  It has been developed under the project leadership of Prof. Dr. Michael. E. Pohst at the University of Duesseldorf from 1987 until 1993 and at TU Berlin afterwards.
  KANT consists of a C-library of thousands of functions (for doing arithmetic). The set of these functions is based on the core of the computer algebra system MAGMA.

# What is KANT/KASH?

# What is KANT/KASH?

- KASH: stands for *KAnt SHell*
  The name of the shell that allows access to the KANT functions. This shell is based on that of the group theory package GAP3 and the handling is similar to that of MAPLE. It makes easier to handle the number theoretical objects.
  The latest version is KASH3.

# Starting and Leaving KASH3

KASH3 is freely available for
Linux/x86, MacOSX and MS Windows from:
*http://www.math.tu-berlin.de/~kant/download.html*

# Starting and Leaving KASH3

KASH3 is freely available for
Linux/x86, MacOSX and MS Windows from:
*http://www.math.tu-berlin.de/~kant/download.html*

- Start by simply typing at the prompt: `kash3`

# Starting and Leaving KASH3

KASH3 is freely available for
Linux/x86, MacOSX and MS Windows from:
*http://www.math.tu-berlin.de/~kant/download.html*

- Start by simply typing at the prompt: `kash3`
- Leave by typing at the prompt: `quit;`

NB: the semicolon is necessary!

# KASH3 - Features

# KASH3 - Features

- Computation in Number Fields: Arithmetic of Algebraic Numbers, Maximal Orders, Integral Bases, Galois Groups up to Degree 23, Unit Groups, Class Groups (unconditionally and under GRH), Class Fields, all Subfields of a Number Field, ...

# KASH3 - Features

- Computation in Number Fields: Arithmetic of Algebraic Numbers, Maximal Orders, Integral Bases, Galois Groups up to Degree 23, Unit Groups, Class Groups (unconditionally and under GRH), Class Fields, all Subfields of a Number Field, ...
- Algebraic Function Fields: Arithmetic of Algebraic Functions, Finite and Infinite Maximal Orders, Primitive elements, Defining Polynomial, Degree, ...

# KASH3 - Features

- Computation in Number Fields: Arithmetic of Algebraic Numbers, Maximal Orders, Integral Bases, Galois Groups up to Degree 23, Unit Groups, Class Groups (unconditionally and under GRH), Class Fields, all Subfields of a Number Field, ...
- Algebraic Function Fields: Arithmetic of Algebraic Functions, Finite and Infinite Maximal Orders, Primitive elements, Defining Polynomial, Degree, ...
- Local Fields: p-adic Fields, IsSquare, IsUnit, Polynomial Factorization, ...

# KASH3 - Features

- Computation in Number Fields: Arithmetic of Algebraic Numbers, Maximal Orders, Integral Bases, Galois Groups up to Degree 23, Unit Groups, Class Groups (unconditionally and under GRH), Class Fields, all Subfields of a Number Field, ...
- Algebraic Function Fields: Arithmetic of Algebraic Functions, Finite and Infinite Maximal Orders, Primitive elements, Defining Polynomial, Degree, ...
- Local Fields: p-adic Fields, IsSquare, IsUnit, Polynomial Factorization, ...
- Diophantine Equations: Norm Equations, Thue Equations, Unit Equations, ...

# KASH3 - Type System

Three different kinds of Types:

# KASH3 - Type System

Three different kinds of Types:

- Simple Types (level 0): stand for themselves

# KASH3 - Type System

Three different kinds of Types:

- Simple Types (level 0): stand for themselves
  - Examples: `list, set, dry, func, ...`
    $[2, 8, , , , "aha"]$; $Set([3, 1, 9, , , 1, 7])$;
    $Dry([5, 4, "string", , , 5, 1])$;

# KASH3 - Type System

Three different kinds of Types:

- Simple Types (level 0): stand for themselves
  - Examples: `list, set, dry, func, ...`
    $[2, 8, , , , "aha"]$; $Set([3, 1, 9, , , 1, 7])$;
    $Dry([5, 4, "string", , , 5, 1])$;
- (Typed) Aggregate Types: are written as

# KASH3 - Type System

Three different kinds of Types:

- Simple Types (level 0): stand for themselves
  - Examples: `list, set, dry, func, ...`
    $[2, 8, , , , "aha"]$; $Set([3, 1, 9, , , 1, 7])$;
    $Dry([5, 4, "string", , , 5, 1])$;
- (Typed) Aggregate Types: are written as
  - `<aggtype>([<type>,<type>])`

# KASH3 - Type System

Three different kinds of Types:

- Simple Types (level 0): stand for themselves
  - Examples: `list, set, dry, func, ...`
    $[2, 8, , , , "aha"]$; $Set([3, 1, 9, , , 1, 7])$;
    $Dry([5, 4, "string", , , 5, 1])$;
- (Typed) Aggregate Types: are written as
  - `<aggtype>([<type>,<type>])`
  - Examples: `map(),seq(),tup()`
    $Sequence(["abd3", "yes", "no3"])$;
    $Tuple([3, "OK", 7.9])$;
    `f:= Map(R, C, x->x*I);`

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$

    atoms for algebraic structure: $alg$, $fld$, $ord$, ...
    atoms for specifier: $fun$, $num$, $pol$, $pad$, ...

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$
    atoms for algebraic structure: $alg$, $fld$, $ord$, ...
    atoms for specifier: $fun$, $num$, $pol$, $pad$, ...
  - Examples:

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$
    atoms for algebraic structure: $alg$, $fld$, $ord$, ...
    atoms for specifier: $fun$, $num$, $pol$, $pad$, ...
  - Examples:
    - $fld\hat{}fun$: Type of Function Fields

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$
    atoms for algebraic structure: $alg$, $fld$, $ord$, ...
    atoms for specifier: $fun$, $num$, $pol$, $pad$, ...
  - Examples:
    - $fld\hat{}fun$: Type of Function Fields
    - $ord\hat{}num$: Type of Orders of Number Fields

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$
    atoms for algebraic structure: $alg$, $fld$, $ord$, ...
    atoms for specifier: $fun$, $num$, $pol$, $pad$, ...
  - Examples:
    - $fld\hat{}fun$: Type of Function Fields
    - $ord\hat{}num$: Type of Orders of Number Fields
    - $elt - ord\hat{}num$: Type of Algebraic Integers

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$
    atoms for algebraic structure: $alg$, $fld$, $ord$, ...
    atoms for specifier: $fun$, $num$, $pol$, $pad$, ...
  - Examples:
    - $fld\hat{}fun$: Type of Function Fields
    - $ord\hat{}num$: Type of Orders of Number Fields
    - $elt - ord\hat{}num$: Type of Algebraic Integers
    - $alg\hat{}pol/fld\hat{}num$: Type of Polynomial Algebras over Number Fields

# KASH3 - Type System

- Composed Types are composed from several atoms and they written as:
  - $[elt-](algebraic\ \ structure)\hat{}(specifier)[/(...)]$
    atoms for algebraic structure: $alg$, $fld$, $ord$, ...
    atoms for specifier: $fun$, $num$, $pol$, $pad$, ...
  - Examples:
    - $fld\hat{}fun$: Type of Function Fields
    - $ord\hat{}num$: Type of Orders of Number Fields
    - $elt - ord\hat{}num$: Type of Algebraic Integers
    - $alg\hat{}pol/fld\hat{}num$: Type of Polynomial Algebras over Number Fields
    - $elt - alg\hat{}mat/fld\hat{}rat$: Type of Matrices with Rational Coefficients

# KASH3 - Type System

Exercises:

1) The type of complex numbers and the complex field?
2) How to install the new type `elt-case^num`?

# KASH3 - Type System

- Useful Functions:

  `ShowTypes();`

  displays a list of all type atoms by level.

  `?*.|TYPE`

  shows all types in KASH3.

  `NewType(newtype,level);`

  installs the type "newtype" of level "level".

Exercises:

1) The type of complex numbers and the complex field?
2) How to install the new type `elt-case^num`?

# KASH3 - Help System

For more information, see:
*http://www.math.tu-berlin.de/~kant/doc.html*

# KASH3 - Help System

- On-line documentation

For more information, see:
*http://www.math.tu-berlin.de/~kant/doc.html*

# KASH3 - Help System

- On-line documentation
- Help system written in the shell

For more information, see:
*http://www.math.tu-berlin.de/~kant/doc.html*

# KASH3 - Help System

- On-line documentation
- Help system written in the shell
- Handbooks and tutorials are compiled from documentation records

For more information, see:
*http://www.math.tu-berlin.de/~kant/doc.html*

# KASH3 - Help System

- On-line documentation
- Help system written in the shell
- Handbooks and tutorials are compiled from documentation records
- HTML and print version via XML output

For more information, see:
*http://www.math.tu-berlin.de/~kant/doc.html*

# KASH3 - Help System

- On-line documentation
- Help system written in the shell
- Handbooks and tutorials are compiled from documentation records
- HTML and print version via XML output
- On-line help with complex search patterns

For more information, see:
*http://www.math.tu-berlin.de/~kant/doc.html*

# Operations

There are three kinds of operators in KASH3

NB: the assignment operator is :=

Example: `a1:=7^98796`; How many digits?

# Operations

There are three kinds of operators in KASH3

- Arithmetical operators: `+`, `-`, `*`, `/`, `^`, `mod`.

NB: the assignment operator is :=
Example: `a1:=7^98796`; How many digits?

# Operations

There are three kinds of operators in KASH3

- Arithmetical operators: `+`, `-`, `*`, `/`, `^`, `mod`.
- Comparison operators: `=`, `<`, `>`, `<=`, `>=` `<>`, `in`.
  A comparison result is a boolean value:
  TRUE, FALSE.
  NB: Algebraic elements, ideals, matrices and complex numbers can be compared via `=` and `<>`.

NB: the assignment operator is `:=`
Example: `a1:=7^98796`; How many digits?

# Operations

There are three kinds of operators in KASH3

- Arithmetical operators: `+`, `-`, `*`, `/`, `^`, `mod`.
- Comparison operators: `=`, `<`, `>`, `<=`, `>=` `<>`, `in`.
  A comparison result is a boolean value:
  TRUE, FALSE.
  NB: Algebraic elements, ideals, matrices and complex numbers can be compared via `=` and `<>`.
- Logical operators: Boolean values can be manipulated via logic operators: `not`, `and`, `or`.

NB: the assignment operator is :=
Example: `a1:=7^98796`; How many digits?

# Polynomials

KASH3 can handle multivariate polynomials.
First create the polynomial algebra and then define the polynomial in it.

Example 1:

```
f:= 5*X^7-3*X^4+23;
Evaluate(f,1);
```

Example 2:

```
Qx:=PolynomialAlgebra(Q);
x:=Qx.1;
AssignNames_(Qx,["x"]);
Qxy:=PolynomialAlgebra(Qx);
y:=Qxy.1;
AssignNames_(Qxy,["y"]);
Hxy:=x^4+5*x*y^3-7*y^2+x*y+2;
```

# Polynomials

Exercises:

1) Define a polynomial with coefficients in $\mathbb{Q}$. Evaluate it at $2$, factorize it.

2) Find a function in KASH3 to compute the cyclotomic polynomial of degree $18$ (the roots are $27^{th}$ roots of the unit).

3) Compute the Sylvester matrix of $2$ different polynomials ($\neq 0$) with coefficients in $\mathbb{Q}$. Compute the determinant of this matrix and compare it to the resultant of the both polynomials.

4) Compute and factorize:

```
-x^2-x*y+x*z+y*z
```

# Kant Database

QaoS: Query algebraic objects System

*http://www.math.tu-berlin.de/~kant/database.html*

# Kant Database

QaoS: Query algebraic objects System

*http://www.math.tu-berlin.de/~kant/database.html*

- Database for Field Extensions

contains 1.3 million number fields

# Kant Database

QaoS: Query algebraic objects System

*http://www.math.tu-berlin.de/~kant/database.html*

- Database for Field Extensions
  - Algebraic Extensions

contains 1.3 million number fields

# Kant Database

QaoS: Query algebraic objects System

*http://www.math.tu-berlin.de/~kant/database.html*

- Database for Field Extensions
  - Algebraic Extensions
  - Transcendental Extensions

contains 1.3 million number fields

# Kant Database

QaoS: Query algebraic objects System

*http://www.math.tu-berlin.de/~kant/database.html*

- Database for Field Extensions
  - Algebraic Extensions
  - Transcendental Extensions

  contains 1.3 million number fields
- Database for Transitive groups

  All the 40226 transitive groups of degree up to 30

# Kant Database

QaoS: Query algebraic objects System

*http://www.math.tu-berlin.de/~kant/database.html*

- Database for Field Extensions
  - Algebraic Extensions
  - Transcendental Extensions

  contains 1.3 million number fields
- Database for Transitive groups
  All the 40226 transitive groups of degree up to 30
- Access from
  GAP4, KASH2.5, KASH2.6, KASH3, Maple and webbrowser