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

Applications

Applications

- Residue Class Ring

Applications

- Residue Class Ring
- Complex Numbers

Applications

- Residue Class Ring
- Complex Numbers
- Matrices

Applications

- Residue Class Ring
- Complex Numbers
- Matrices
- Lattices

Applications

- Residue Class Ring
- Complex Numbers
- Matrices
- Lattices
- Number Fields

Applications

- Residue Class Ring
- Complex Numbers
- Matrices
- Lattices
- Number Fields
- Local Fields

Applications

- Residue Class Ring
- Complex Numbers
- Matrices
- Lattices
- Number Fields
- Local Fields
- Ideals

Applications

- Residue Class Ring
- Complex Numbers
- Matrices
- Lattices
- Number Fields
- Local Fields
- Ideals
- Programming Language

Residue Class Ring

Create the residue class ring $\mathbb{Z}/m\mathbb{Z}$ using the function `ResidueClassRing` or `IntegerRing`.

Example 1:

```
A:=ResidueClassRing(6);
```

NB: We use the function `Element` or `Coerce` to coerce an element in a set.

Example 2:

```
Element(ResidueClassRing(5),16543);
```

Some Functions: `MultiplicativeGroup`, `Size`, `PrimitiveElement`, `AdditiveGroup`, ...

Residue Class Ring

Exercises:

1) Find a function to compute all the square roots of the unity in the residue class ring of integers $modulo$ 9.

2) Compute the inverse of 23467879 and 765432198673 in the residue class ring of integers $modulo$ 11. Compute all the units.

3) Compute the order of 6 in the residue class ring of integers $modulo$ 7^7.

Complex Numbers

Complex Numbers have a default precision of 30.
One can change the precision to arbitrary n.
The complex number `I` such that `I^2=-1` is predefined in KASH3.
Examples: `z:= 3+2*I; z^3`; `z^1/8`;
`Exp(2*PI*I); (2+9*I)/(1+5*I);`

NB: most real functions can be applied to complex numbers.

Exercises: $1)$ Using the relation `Gamma(z+1)=z*Gamma(z)` $\forall z \in \mathbb{C}$ such that $Re(z) > 0$, show that `Gamma(n)` is a positive integer $\forall n \in \mathbb{N}\backslash\{0\}$.
$2)$ Give the polar form of: `4-7*I; 1/(3+I);`

Matrices

First give the coefficients ring.
Then the number of rows and columns and finitely a list consisting of the entries.

Example 1: $M :=$
`Matrix`$(Z, 5, 3, [2, 4, 7, 8, 9, 3, 4, 6, 5, 2, 1, 6, 8, 4, 3])$;

Remark: It is not necessary to define the ground ring.

Example 2:

$N :=$ `Matrix`$(2, 3, [2, 4, 7, 8, 9, 3])$;

$P :=$
`Matrix`$(4, [2, 4/5, 1, 58, 9, 13, 0, 54, 8, 8, 1, 0, 2, 7, 1, 7])$;

Some functions:
`KernelMatrix,Transpose,SmithForm,Adjoint`
`GramMatrix,IsUnipotent,Determinant`

Matrices

Exercises:

1) the smith normal form of
$A := Matrix(Z, 3, 3, [2, 4, 4, -6, 6, 12, 10, -4, -16])$;
Is A invertible? Compute its adjoint, its Gram matrix and its eigen values. Compute the smith normal form of A.

2) Given the matrix $M :=$
`Matrix`$(Z, 5, 3, [2, 4, 7, 8, 9, 3, 4, 6, 5, 2, 1, 6, 8, 4, 3])$;
Compute 2 unimodular square matrices `P` and `T` such that `P*M*T=S` where $S :=$
$Matrix(Z, 5, 3, [1, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0])$;

3) Create a square matrix 5×5 with coefficients in $\mathbb{Q}$ and find a function in KASH3 to compute its determinant. Compute its inverse if it is invertible.

Lattices

In KASH3 lattices are represented as matrices. We can compute the Gram matrix and a LLL-reduced basis of a lattice.

`LLL` applied to a matrix M returns a matrix L whose rows are a LLL reduced basis for the lattice (over a real subring) spanned by the rows of M together with a unimodular matrix T over $\mathbb{Z}$ such that L = T * M, and the rank of M.

Also, we get `LLLGram`: it returns a LLL-reduced form of the Gram matrix M, together with the corresponding transformation matrix T and the rank of M.

Lattices

Examples:

```
N:=Matrix(Q,3,[1/2,3,2,3,0,1,2,9/2,2]);
LLL(N);
A:=GramMatrix(N);
LLLGram(A);
```

Exercises:

1) Compute a square matrix 4×4, named A, with coefficients in $\mathbb{Q}$. Compute a unimodular matrix B such that $B.A$ has its rows LLL reduced basis for a lattice over a real subring. Give the rank of A.

2) Is it possible to compute the `LLLGram` matrix of `M:=Matrix`$(4, [1/2, 3, 2, 5, 3, 7, 8, 9, 2, 8, 5, 6, 5, 9, 6, 2])$;

Number Fields

A finite extension of the field of rational numbers $\mathbb{Q}$
It is generated by the root of a monic irreducible polynomial with coefficients in $\mathbb{Z}$.

Examples:

```
NumberField(X^2+2);
NumberField(X^8+7*X^5+1);
```

Some Computations in number fields:

```
Subfields,IsSubfield,EquationOrder,
MaximalOrder,Galois,Basis,UnitGroup,
ClassGroup,ClassNumber,UnitRank,
PrimitiveElement,RingOfIntegers
```

Number Fields

Exercise:

Compute the number field K generated by the polynomial:

```
f:=X^9-3*X^6-9*X^3+3;
```

Compute a primitive element of K and a basis of K.

Compute the ring of integers of K and a basis of this ring.

Compute the discriminant d_K of K (the discriminant of the ring of integers of K) and the discriminant d_f of `f`.

Apply `IsSquare` to d_f/d_K. Conclusion.

Compute the Galois group of K (the Galois group of the generating polynomial).

Local Fields

KASH3 can handle p-adic rings and p-adic fields
Examples: $\mathbb{Z}_5$, $\mathbb{Q}_7$

`pAdicRing(3);` give the 3-adic ring $\mathbb{Z}_3$
`pAdicRing(3,6);` give the 3-adic ring $mod\, 3\hat{}6$
`pAdicField(11);` give the 11-adic field $\mathbb{Q}_{11}$
`pAdicField(11,8);` give the 11-adic field $mod\, 11\hat{}8$

Some Computations in Local Fields:

`pAdicRing,pAdicField,LaurentSeriesRing,`
`DefiningPolynomial,ResidueClassField,`
`TotallyRamifiedExtension,Factorization,`
`UniformizingElement`

Local Fields

Exercise:
Compute the polynomial `f:=Y^3+626` with coefficients in the 5-adic ring `mod` `5^4`. Factorize it.

Ideals

KASH3 can handle integral ideals and fractional ideals.

Example 1:

```
L:=Ideal(Z,5);
J:=Ideal(EquationOrder(X^2+1),[7,123]);
K:=(1/2)*Ideal(MaximalOrder(X^3+2),4,9);
```

Example 2:

```
M:=MaximalOrder(X^2+5);
N:=Ideal(M,Matrix(Z,2,2,[1,1,0,2]));
ResidueClassField(N);
```

Some Functions:

```
Intersection,IsPrime,AbsoluteNorm,
Divisors,Degree, InertiaDegree,
IsPrincipal,RamificationDegree,
MakeCoprime,BasisMatrix, ...
```

Ideals

The dictionary between fractional ideals and rational numbers:

Integral ideals	$\leftrightarrow$	integers
Fractional ideals	$\leftrightarrow$	(non zero) rational numbers
Inclusion	$\leftrightarrow$	divisibility
Sum	$\leftrightarrow$	GCD
Intersection	$\leftrightarrow$	LCM
Product	$\leftrightarrow$	product

NB:

If $(I_i)_{i \in J}$ are pairwise coprime ideals then $\bigcap_{i \in J} I_i = \prod_{j \in J} I_j$.

Ideals

Exercises:

1) `Ideal(Z,6,9);`

2) Compute the intersection, the sum and the product of

   a) `12Z` and `9Z`.

   b) `5Z` and `7Z`.

3) Compute the maximal order `A` of `f:=X^2+5`.

Compute the ideal `J` generated by 27 and 33 in `A`. Is it a principal ideal? Give the generator. Give the divisors of `J`. Is it possible to coumpute the residue class field of `J`? Is `A` a PID?

Programming Language

KASH3 uses the GAP3 shell as a user interface. The programming language of GAP3 is an imperative language with some functional and some objects oriented features. In KASH3 additional features like Methods, Maps, and Extendable Objects are available.

function

Synthax:

$function([< arg-ident >,< arg-ident >])$
$[local < loc-ident >,< loc-ident >;]$
$< statements >$
$end;$

Purpose:

A function is in fact a literal and not a statement; so it can be assigned to a variable or to a list element or a record component.
It is possible that a function calls itself, this is usually called recursion.
A function <fun1> definition can be evaluated inside another function <fun2>.

function:

Example:

```
kash% addition:= function(arg1, arg2)
% #this function returns the sum of
% #the both arguments
% local a;
% a:= arg1+arg2;
% Print("The sum is:\n");
% return a;
% end;
```

function

NB:
A comfortable way to define a "simple" function is to use the maps-to operator: `->`
Examples:

```
cube:= x -> x^3;
cube(3)? cube(6.9)? cube(I)?
M:=Matrix(3,[2,8,9,5,4,0,1,3,2]);
cube(M)?

additionby5:= x -> x+5;
additionby5(0)? additionby5(-76)?
```

for

Synthax:

$for < variable > in < list > do < statements > od;$

Purpose:

The for loop executes the <statements> for every element of <list>. The statement sequence <variable> is first executed with <variable> bound to the first element of <list>, then with <variable> bound to the second element of <list> and so on. <variable> must be a simple variable, it must not be a list element selection or a record component selection.

for

Example:

```
kash% changelist:= function(L)
% #this function takes a list as
% #argument and changes its entries
% local i, K;
% K:= []; K[1]:= L[1];
% L[1]:= K[1]-2*L[Length(L)];
% for i in [2..Length(L)] do
%  K[i]:= L[i]; L[i]:= K[i]-2*K[i-1];
% od;
% return L;
% end;
```

What does it do?

if

Synthax:

$if < elt - alg\hat{}boo > then < statements1 >;$
$\{elif < elt - alg\hat{}boo > then < statements2 >\}$
$[else < statements3 >]$
$fi;$

Purpose:

The if statement allows one to execute statements depending on the value of some boolean expression. The if statement terminates by a fi keyword.

if

Example:

```
kash% checkprimenumber:= function(a)
% #this function checks if the given
% #number is prime or not!
% if IsPrime(a) then
%    Print(a);
%    Print(" is a prime number \n");
% else
%    Print(a);
%    Print(" is not a prime, bye!\n");
% fi;
% end;
```

while

synthax:

$$while < elt - alg\hat{}boo > do < statements > od;$$

Purpose:

The while loop executes the <statements> while the condition evaluates to true. First the boolean expression is evaluated. If it evaluates to false execution of the while loop terminates and the statement immediately following the while loop is executed next. Otherwise if it evaluates to true the <statements> are executed and the whole process begins again.

repeat

Synthax:

$repeat < statements > until < elt - alg\hat{}boo >;$

Purpose:

The repeat loop executes the statement sequence <statements> until the condition evaluates to true. First <statements> are executed. Then the boolean expression is evaluated. If it evaluates to true the repeat loop terminates and the statement immediately following the repeat loop is executed next. Otherwise if it evaluates to false the whole process begins again with the execution of the <statements>.

Examples

Example 1

```
kash% Mul_and_Inv:= function(arg1)
% #this function returns a map that
% #multiplies by "arg1"
% local phi, psi;
% phi:= function(arg2)
%   return arg2*arg1; end;
% psi:= function(arg3)
%   return arg3/arg1; end;
% return Map(Q, Q, phi, psi);
% end;
```

Remarque

The keyword `arg` is predefined in KASH3.

`function(arg) ... end;`

Collapse arbitrary many arguments to a list of arguments and pass it to `arg`.

Example:

```
kash%
x_f:= function(arg) return arg; end;
kash% x_f(1, 5.8, "yes", [3,45])?
x_f(7)?
```

Examples

Example 2

```
kash% gcd_int:= function(arg)
% #this function returns the GCD
% #of the given integers.
% local c, r, how, i;
% if Length(arg)=1 then return arg[1];
% elif Length(arg) <> 1 then
%   how:= function(a, b)
%    while b <> 0 do
%    r:= b; b:= a mod b; a:= r; od;
%    return a;
%   end;
%   c:= how(arg[1], arg[2]);
%   for i in [3..Length(arg)] do
```

Examples

```
%   c:= how(c, arg[i]); od;
%   return c;
% fi;
% end;
```

Examples

Example 3

```
kash% GCD_int:= function(arg)
% #this function returns the GCD
% #of the given integers.
% local c, r, how, i;
% if Length(arg)=1 then return arg[1];
% elif Length(arg) <> 1 then
%   how:= function(a, b)
%    repeat r:= a mod b; a:= b; b:= r;
%    until b=0;
%    return a;
%   end;
%   c:= how(arg[1], arg[2]);
%   for i in [3..Length(arg)] do
```

Examples

```
%   c:= how(c, arg[i]); od;
%   return c;
% fi;
% end;
```

Examples

Example 4: The program file prog.k

```
InverseMatrix:= function(M)
#this function returns the inverse
#of a matrix over Z if it is
#invertible using the formula:
#Inv(M) = Adjoint(M)*(1/Det(M))
local A;
A:= Determinant(M);
if BaseRing(M) <> Z then
return "The coeff are not in Z,bye!";
elif A = -1 or A = 1 then
 Print("The inverse is:  \n");
 Print(Adjoint(M)*(1/A), "\n");
else
```

Examples

```
  Print("The determinant is:  ");
  Print(Determinant(M), "\n");
  Print("Not invertible over Z! \n");
fi;
end;
```

How to load the program file prog.k in KASH?

```
kash%  Read("prog.k");
```