AGNES Scavenger Hunt with Magma

Eran Assaf, Edgar Costa

November 2024

If you encounter any problem, please share it with us via the form `https://forms.gle/52pCHhbKofi3DvMP7`.

1 Quick overview of Magma

1.1 What is Magma?

`Magma` is a computer algebra system for computations in algebra, number theory, and geometry, as described in [3]. We strive to provide:

- a mathematically rigorous environment, with
- highly efficient algorithms and implementations.

Magma consists of a large C kernel to achieve efficiency, and an ever increasing package of library functions programmed in the Magma language for higher level functions.
`https://magma.maths.usyd.edu.au/magma/handbook/`

1.2 The Magma model

The Magma model is based on concepts from category theory.

- Every object belongs to a (unique, extended) **category**, a class of objects belonging to a **variety** that share the same representation.

```
> a := [1];
> Category(a);
SeqEnum
> ExtendedCategory(a);
SeqEnum[RngIntElt]
```

- Every object has a (unique) **parent** structure, describing the mathematical context in which it is viewed.

```
> Parent(a);
Set of sequences over Integer Ring
> Parent(a[1]);
Integer Ring
```

- Univariate and multivariate polynomial rings form the categories `RngUPol` and `RngMPol` (which lie in the variety `Rng`). Univariate polynomials over the integers form an **extended** category `RngUPol[RngInt]`. The parent of a univariate polynomial is its polynomial ring.

```
> k := Rationals();
> kt<t> := PolynomialRing(k);
> kt;
Univariate Polynomial Ring in t over Rational Field
> Category(kt);
RngUPol
> ISA(RngUPol, Rng);
true
> ExtendedCategory(kt);
RngUPol[FldRat]
> kx<[x]> := PolynomialRing(k, 10);
> kx;
Polynomial ring of rank 10 over Rational Field
Order: Lexicographical
Variables: x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10]
> Category(kx);
RngMPol
> ISA(RngMPol, Rng);
true
```

1.3 Acknowledgments

We would like to thank Asher Auel and John Voight for helpful discussions, Juanita Duque-Rosero, Pim Spelier and John Voight for organizing similar activities from which we drew inspiration, and Avi Kulkarni and Andrew Sutherland for other helpful tips.

2 First Steps

This section is intended for applications in Algebraic Geometry. For a more comprehensive set of examples, one may refer the Magma Handbook or the excellent guide [1].

2.1 Schemes

In this example we write down a monomial scheme and analyze the basic properties of its components. We first construct $\mathbb{A}^3_{\mathbb{Q}}$, the affine 3-space over the rationals.

```
> A<x,y,z> := AffineSpace(Rationals(),3);
> A;
Affine Space of dimension 3 over Rational Field
Variables: x, y, z
```

Define the subscheme of $\mathbb{A}^3_{\mathbb{Q}}$ defined by a sequence of polynomials $X = \operatorname{Spec} k[x,y,z]/(xy^2, x^4z)$.

```
> X := Scheme(A,[x*y^2,x^4*z]);
> X;
Scheme over Rational Field defined by
x*y^2,
x^4*z
```

We find that $\dim X = 2$, it is nonreduced, and find its primary components.

```
> Dimension(X);
2
```

```
> IsReduced(X);
false
> Xprim := PrimaryComponents(X);
```

`Xprim` is a sequence of components; to access elements, use brackets.

```
> Xprim[1];
Scheme over Rational Field defined by
x
> Xred, f := ReducedSubscheme(X);
> Xred;
Scheme over Rational Field defined by
x*y,
x*z
```

Comparison in `Magma` is `eq`.

```
> AmbientSpace(X) eq A;
true
> I := DefiningIdeal(X);
> I;
Ideal of Polynomial ring of rank 3 over Rational Field
Order: Lexicographical
Variables: x, y, z
Homogeneous, Dimension 2, Non-radical, Non-primary, Non-prime
Basis:
[
    x*y^2,
    x^4*z
]
```

The ring containing I.

```
> R := Generic(I);
> R;
Polynomial ring of rank 3 over Rational Field
Order: Lexicographical
Variables: x, y, z
```

Define a subscheme of projective space.

```
> Proj(R/I);
Scheme over Rational Field defined by
x*y^2,
x^4*z
Projective Space of dimension 2 over Rational Field
Variables: x, y, z
```

2.2 Groebner Bases

We compute the Groebner basis of the "Cyclic-6" ideal with respect to the lexicographical order.

```
> P<x,y,z,t,u,v> := PolynomialRing(Rationals(), 6);
```

```
> sigma := hom<P -> P | [y,z,t,u,v,x]>;
> sigma(x);
y
> gens := [ &+[ (sigma^i)(m) : i in [1..6] ] :
>              m in [x,x*y,x*y*z,x*y*z*t,x*y*z*t*u] ] cat [x*y*z*t*u*v - 1];
> I := ideal<P | gens>;
```

Compute a Groebner basis.

```
> time gb := GroebnerBasis(I);
Time: 0.070
> #gb;
17
```

Now we can compute dimension.

```
> Dimension(I);
0 []
```

Since the ideal is zero-dimensional and the monomial order is lex, the last polynomial in the lex Groebner basis is univariate.

```
> gb[17];
v^48 - 2554*v^42 - 399710*v^36 - 499722*v^30 + 499722*v^18 + 399710*v^12 + 2554*v^6 - 1
> time Factorization(gb[17]);
[
    <v - 1, 1>,
    <v + 1, 1>,
    <v^2 + 1, 1>,
    <v^2 - 4*v + 1, 1>,
    <v^2 - v + 1, 1>,
    <v^2 + v + 1, 1>,
    <v^2 + 4*v + 1, 1>,
    <v^4 - v^2 + 1, 1>,
    <v^4 - 4*v^3 + 15*v^2 - 4*v + 1, 1>,
    <v^4 + 4*v^3 + 15*v^2 + 4*v + 1, 1>,
    <v^8 + 4*v^6 - 6*v^4 + 4*v^2 + 1, 1>,
    <v^8 - 6*v^7 + 16*v^6 - 24*v^5 + 27*v^4 - 24*v^3 + 16*v^2 - 6*v + 1, 1>,
    <v^8 + 6*v^7 + 16*v^6 + 24*v^5 + 27*v^4 + 24*v^3 + 16*v^2 + 6*v + 1, 1>
]
Time: 0.000
```

More generally, we can write a short function for creating the cyclic n-ideal.

```
function CyclicNIdeal(n);
  P := PolynomialRing(Rationals(), n);
  sigma := hom<P -> P | [P.i : i in [2..n] cat [1]]>;
  monoms := [P.1];
  for i := 2 to n do
    Append(~monoms, monoms[i-1]*P.i);
  end for;
  gens := [ &+[ (sigma^i)(m) : i in [1..n] ] : m in monoms[1..n-1] ]
     cat [ monoms[n] - 1];
  return ideal<P | gens>;
end function;
```

We can instruct the Faugere algorithm to be chatty so we can watch the progression of the computation.

```
> SetVerbose("Groebner", 1);
> time gb := GroebnerBasis(CyclicNIdeal(7));
Homogeneous weights search
Number of variables: 7, nullity: 1
Exact search time: 0.000
Modular Groebner Basis over Q
Rank: 7
Order: Graded Reverse Lexicographical
Module col limit: -1
Pairs limit: -1
Degree limit: -1

Step 1 (try 1), basep: 1, nr: 1, 0.000
GB time: 0.289[r]
Is zero dimensional: 1

Step 2 (try 2), basep: 2, nr: 1, 0.290
GB time: 0.220
Step 3 (try 3), basep: 3, nr: 1, 0.560
GB time: 0.067[r]
Final GB length: 209
Total number of steps: 3
Total modular partial GB time: 0.650
Switch to standard F4
********************
FAUGERE F4 ALGORITHM
...
New polynomial 34, leading monomial: $.1, 7.558[r]
Total FGLM time: 7.580
Time: 8.980
...
```

Magma can work with many different kinds of term orders and ground rings.

```
> Z := IntegerRing();
> P<a,b,c> := PolynomialRing(Z, 3, "weight", [4,2,1, 1,2,3, 1,1,1]);
```

2.3 Resolution graph of a curve singularity

Let $C = \operatorname{Spec} \mathbb{Q}[x,y]/((x^2 - y^3)^2 + xy^6)$ be a singular affine curve.

```
> A<x,y> := AffineSpace(Rationals(),2);
> C := Curve(A,(x^2 - y^3)^2 + x*y^6);
> C;
Curve over Rational Field defined by
x^4 - 2*x^2*y^3 + x*y^6 + y^6
```

The interesting singularity is at the origin. We calculate its resolution graph and display it.

```
> g := ResolutionGraph(C,Origin(A));
> g;
The resolution graph on the Digraph
```

```
Vertex Neighbours

1 ([ -3, 4, 1, 0 ]) 2 ;
2 ([ -2, 12, 4, 0 ]) 3 4 ;
3 ([ -2, 6, 2, 0 ]) ;
4 ([ -3, 14, 5, 0 ]) 5 ;
5 ([ -1, 30, 12, 1 ]) 6 ;
6 ([ -2, 15, 6, 0 ]) ;
```

The resulting graph has 6 vertices. So the transverse resolution of this singularity is achieved by 6 blowups. In brackets, each vertex has a label of the form $[s, m, k, t]$, where:

- s is the self-intersection;
- m is the multiplicity (dependent on context);
- k is the canonical multiplicity;
- t is the number of transverse intersections at v.

```
> Genus(C);
0
```

3 Helpful tips

This section is an adaptation of [2].

Some useful links: Documentation, general examples.

3.1 How do I get help?

- To access the documentation of a function, write `FunctionName;`.
- You can write the first letter of a function and tab complete twice to get a list of possible functions.
- You can try `ListSignatures(Type(YourElement));` to list all `Magma` functions that accept `YourElement` as input.
- If there are multiple functions with name `FunctionName`, you can access the ones relevant to `YourElement` by typing the following commmand - `ListSignatures(FunctionName, Type(YourElement));`.

3.2 Warning!

- Every command must end with a semicolon. Nothing happens until Magma sees the semicolon.
- `Magma` is fussy about types (curves and schemes are different types, for example) and elements may need to be explicitly coerced to have the type you want. To change the scheme C to a curve, use `Curve(C)`. Similarly, to change a scheme S to a surface, use `Surface(Ambient(S), Ideal(S))`.
- When defining a map between schemes $f : X \to Y$ using polynomials $p_1, \ldots, p_n$, `Magma` would create a rational map even if the polynomials have a base scheme over which they all vanish, i.e. $V = V(p_1, \ldots, p_n)$ is nonempty. You may check whether this is an honest morphism of schemes using `IsMorphism`.

3.3 Other useful tips

- To load a file, use

```
> load "fileAddress";
```

- To kill a process, use `control + c`. To kill Magma, do this twice within a second.
- To ignore `>`, use the command `SetIgnorePrompt(true);`. This is very helpful when you are copying code from, let's say, worksheets.
- `$1` denotes the last printed result (you can also call `$2` and `$3`). More dollar values can be made available by using, for example, `SetPreviousSize(10);`.
- When you are debugging functions, you can `SetDebugOnError(true);`. This will give you access to the "inside" of your function, up to where `Magma` got stuck. You print things by `p whateverYouWant`. Use `q` to quit back to the `Magma` terminal. You can see all the functions that were used in your computation using `bt` (shows you the "frames"). Go to a specific frame by writing `f theNumberYouWant`. Warning: do not use ;.
- In your home directory you can create a file `.magmarc`, if does not exist, to have certain commands to run on start. For example, you can set `QQ := Rationals();`
- Use `control + e` to get to the last character of a line in the terminal. Use `control + a` for the first one. Do `control + k` to delete the line. You can also find other combinations here. If you are used to editing with `vi`, the command `SetViMode(true);` could be useful.
- Use `%P` to print the Magma input so far, or `%p` to do the same with line numbers shown.
- You can search only for signatures without inheritance:

  `ListSignatures(Srfc : Isa:=false);`. Moreover, you can also just look for functions where your type is an argument or a return values (very useful when you try to find a function producing the type that another function needs. . .):

  `ListSignatures(Crv : Search:="ReturnValues", Isa:=false);`.

4 Scavenger Hunt

We will find a positive integer n.

1. Let E be the elliptic curve over $\mathbb{Q}$ given by the Weierstrass equation $y^2+y = x^3-x^2$. Find all rational points on E. Let $n_1 = \#E(\mathbb{Q})$.
2. Let $S \subseteq \mathbb{P}^4$ be the surface defined by $\sum_{i=0}^4 x_i = 0 = \sum_{i=0}^4 x_i^3$. Compute n_2, the number of rational points on S where 3 coplanar lines of the surface meet.

   [The command `EckardtPoints` may be useful. Note that it only takes as input Del Pezzo surfaces, so it might be useful to use `IsDelPezzo`. For more information on Del Pezzo surfaces, refer to `https://magma.maths.usyd.edu.au/magma/handbook/text/1439`]
3. Let $X_1, X_2 \subseteq \mathbb{P}^4_{\mathbb{Q}}$ be the 2-planes defined by the following equations.

$$X_1 : 6x_1 - 8x_3 + 10x_4 - 5x_5 = 2x_1 + 10x_2 + 8x_3 + 8x_4 + x_5 = 0$$
$$X_2 : 5x_1 + 9x_2 + x_4 = 5x_1 + 5x_2 + 4x_3 + 8x_5 = 0$$

   Let $X = X_1 \cup X_2$. Verify that X is not Cohen-Macaulay. Find $n_3 = h^1(X, \mathscr{O}_X(-10))$.

   [The commands `IsCohenMacaulay`, `StructureSheaf` and `CohomologyDimension` could be useful.]

   In fact, $h^1(X, \mathscr{O}_X(-d)) > 0$ for all $d > 0$, showing that X does not have a dualizing sheaf.

4. Let C be the hyperelliptic curve $y^2 = (x^2-1)(x^2-4)(x^2-9)$, and let $\varphi_5 : C \to \mathbb{P}^8$ be its embedding into projective space by the complete linear series of degree 5. Compute the Betti number $n_4 = \beta_{6,8}(\varphi_5(C))$.

   [Note that `BettiNumber` takes as input a module. One may use `QuotientModule(Ideal(X))` to get a module corresponding to a scheme X.]

   Compute $n = \sum_{i=1}^{4} n_i$ to get your answer.

4.1 Bonus

5. Let $F_n = \mathbb{P}(\mathscr{O}_{\mathbb{P}^1} \oplus \mathscr{O}_{\mathbb{P}^1}(n))$ be the n-th Hirzebruch surface.

   (a) Create F_n via `HirzebruchSurface` as a toric variety (the type is `TorVar`)

   (b) Embed it into projective space. For example, by finding an ample line bundle using `IsProjective`, and take its sections using `RiemannRochBasis`.

   (c) Find a divisor C on F_3 with self-intersection $C^2 = 3$. [There are two natural projections $F_n \to \mathbb{P}^1$ induced by $\mathscr{O}$ and $\mathscr{O}(n)$. Try to construct them explicitly using `pi:=map<Fn->P1 ... >`, and take fibers over a point by using `pt@@pi`.]

   (d) Compute $n_5 = h^0(C, \mathscr{O}_C)$, for example, via `DimensionOfGlobalSections`.

   [You might need to find how to convert a divisor into a sheaf, as `DimensionOfGlobalSections` take as input a sheaf (type `ShfCoh`).]

6. Consider the following K3 surface, over given as a degree 2 cover of $\mathbb{P}^2$

$$S_\lambda \colon w^2 = x^6 + y^6 + z^6 - \lambda(xyz)^2$$

   (a) We say that S_λ has a tritangent line if there is a line in the plane that splits in S_λ. This condition is equivalent to the line l meeting the ramification sextic with even multiplicity at each intersection point. Find the the two $\lambda \in \mathbb{Q}$ for which S_λ has geometric tritagent lines, but no rational tritangent lines.

   [The command `TriTangentLines` could be your friend, but its source code, which you can find the location by typing `TriTangentLines:Maximal;`, could be better friend if you don't want to do a brute force search. The section "Degree 2 K3 surfaces" of the handbook might be useful `https://magma.maths.usyd.edu.au/magma/handbook/text/1440`]

   (b) n_6 is the sum of the ranks of the groups generated by the lines.

   Compute $n = \sum_{i=1}^{6} n_i$ to get your answer.

References

[1] Wieb Bosma, John Cannon, Catherine Playoust, and Allan Steel, *Solving problems with MAGMA*, 1999.

[2] Juanita Duque-Rosero, *Introduction to Computations in Magma*, 2022.

[3] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265, DOI 10.1006/jsco.1996.0125. Computational algebra and number theory (London, 1993).