Tame Genus Package

Joshua Maglione

Univeristät Bielefeld
jmaglione@math.uni-bielefeld.de

Peter A. Brooksbank

Bucknell University
pbrooksb@bucknell.edu

James B. Wilson

Colorado State University
James.Wilson@ColoState.Edu

Version 2.0
November 25, 2019

Contents

CHAPTER 1

Introduction

The goal of this package is to provide MAGMA [**BCP**] with high performance isomorphism tests for groups with tame genus. A group has tame genus if its genus is less than 3. We include automorphism group constructors, along with canonical signatures. For definitions, theorems, and desriptions of algorithms, the reader should consult [**BMW**]. When appropriate, intrinsics for groups also work for tensors (`TenSpcElt`). This package includes a number of intrinsics for automorphism and isomorphism computations, canonical labeling, and constructing random groups of genus 1 or 2 of exponent $\leq p^2$. The index at the end of the documentation records all intrinsics provided by TameGenus.

The latest versions of `TensorSpace` [**MW2**], `StarAlge` [**BW**], and `Sylver` [**MW1**] are required to use this package. The URLs are found in the bibliography or in the package `README.md`. Linux users can use the `install.sh` or `update.sh` bash files to install or update dependencies.

1.1. Citing TameGenus

To cite the TameGenus package, please use the following

Joshua Maglione, Peter A. Brooksbank, and James B. Wilson, *TameGenus*, version 2.0, GitHub, 2019. (https://github.com/thetensor-space/TameGenus).

For AMSRefs:

```
\bib{TameGenusPkg}{misc}{
   author={Maglione, Joshua},
   author={Brooksbank, Peter A.},
   author={Wilson, James B.},
   title={TameGenus},
   publisher={GitHub},
   year={2019},
   edition={version 2.0},
   note={\texttt{https://github.com/thetensor-space/TameGenus}},
}
```

1.2. Verbose Printing and Version

We have included intrinsics to allow for verbose printing.

```
SetVerbose(MonStgElt, RngIntElt) : ->
```

`SetVerbose` is a built in Magma function, but with this package, this intrinsic accepts the string `"TameGenus"` and an integer in $\{0, 1, 2\}$. Verbose printing level 0 turns off all printing. Level 1 will print out what MAGMA is doing in the TameGenus algorithms, and level 2 will print timings and extra structural information.

Example 1.1. `VerbosePrinting`

We demonstrate the verbose printing by constructing a random genus 2 group of order 3^{22}.

```
> G := RandomGenus2Group(3, [4, 6, 10]);
> #G eq 3^(4 + 6 + 10 + 2);
true
```

```
> Genus(G);
2
```

With the verbose printing, we can see how long each part of the algorithm takes and the information it obtains about the input.

```
> SetVerbose("TameGenus", 1);
```

We have included an intrinsic to check which version of TameGenus you have attached in Magma.

`TameGenusVersion() : -> MonStgElt`

Returns the version number for the TameGenus package attached in Magma.

Example 1.2. `Version`

We quickly verify that we have the latest version of TameGenus.

```
```

CHAPTER 2

Constructors

We introduce intrinsics to construct nonabelian groups of genus ≤ 2. We provide some algorithms to construct random groups, and we make no attempt at asserting any analysis of these algorithms. In particular, these random constructors may not select groups in a uniform distribution.

```
TGRandomGroup(q, n, g : parameters) : RngIntElt, RngIntElt, RngIntElt -> GrpPC
    Exponentp : BoolElt : true
```

Given $q = p^m$, $n > 0$, and $g > 0$, returns a p-group G with genus $\leq g$ and order $q^{n+g} = p^{m(n+g)}$. If q is not a prime, then the centroid of the group will be a proper field extension of $\mathbb{F}_p$, containing $\mathbb{F}_q$. If $\circ : V \times V \rightarrowtail W$ is the commutation tensor of G, then $\dim_{\mathcal{C}(\circ)}(V) \leq n$ and $\dim_{\mathcal{C}(\circ)}(W) \leq g$, where $\mathcal{C}(\circ)$ is the centroid of $\circ$. The centroid may strictly contain $\mathbb{F}_q$, in which case the dimensions of V and W divide n and g, respectively. It is also possible that the returned group is not directly indecomposable; this may cause other intrinsics to reject such output. The algorithm is based on the Universal Coefficients Theorem, see [**LGM**, Chapter 9] for the statement and proof. There is one optional parameter: `Exponentp`.

`Exponentp`**:** The default is `true`. Set to `false` if returned group need not have exponent p.

```
RandomGenus2Group(q, d : parameters) : RngIntElt, [RngIntElt] -> GrpPC
    Exponentp : BoolElt : true
```

Given $q = p^m$ and $d = [d_1, \ldots, d_k]$, where $d_i > 0$, it returns a genus 2 group G whose commutation tensor $\circ : V \times V \rightarrowtail W$ has centroid $\mathcal{C}(\circ) \cong \mathbb{F}_q$ and whose $\perp$-decomposition has blocks of dimensions $d_1, \ldots, d_k$ (over $\mathcal{C}(\circ)$). If $d_i = 1$, then this increases the dimension of the radical of $\circ$ by 1 (over $\mathcal{C}(\circ)$). There is one optional parameter: `Exponentp`.

`Exponentp`**:** The default is `true`. Set to `false` if returned group need not have exponent p.

```
RandomGenus1Group(q, d, r : parameters) : RngIntElt, RngIntElt, RngIntElt -> GrpPC
    Exponentp : BoolElt : true
```

Given $q = p^m$, $d > 0$, and $r \geq 0$, it returns a group G with genus 1 and of order q^{2d+r+1}. The center of G has order q^{r+1}, and G is $m(2d + r)$-generated. There is one optional parameter: `Exponentp`.

`Exponentp`**:** The default is `true`. Set to `false` if returned group need not have exponent p.

```
Genus2Group(f) : RngUPolElt -> GrpPC
Genus2Group(f) : RngMPolElt -> GrpPC
```

Given either a univariate or homogeneous multivariate polynomial in x (and y) over $\mathbb{F}_q$, it returns a group G whose commutator $\mathbb{F}_q$-tensor has a Pfaffian equivalent to f.

2.1. Direct indecomposability

The definition of genus requires knowing the (direct) indecomposable factors of a group G. Although we do not have intrinsics to produce a list of subgroups that are indecomposable, we have an efficient test to decide if the given group is indecomposable. There is an analog for tensors as well.

```
IsIndecomposable(G) : GrpPC -> BoolElt
IsIndecomposable(t) : TenSpcElt -> BoolElt
```

Given either a group G or a tensor t, decide if the given object is indecomposable.
Warning: We make no claim that the intrinsics in this package will accept input that return `false` from `IsIndecomposable`. There are indeed cases where the group or tensor is not indecomposable (e.g. nontrivial radical) where all the intrinsics will work fine.

2.2. Genus

```
Genus(G) : GrpPC -> RngIntElt
Genus(T) : TenSpcElt -> RngIntElt
```

Given an indecomposable p-group G, it returns the genus of G. That is, the rank of $[G, G]$ over the centroid. For an indecomposable tensor $t : V \times V \rightarrowtail W$, it returns the rank of W over its centroid.

CHAPTER 3

Isomorphism

We have two intrinsics for deciding isomorphism: one for groups and one for tensors.

```
TGIsIsomorphic(G, H : parameters) : GrpPC, GrpPC -> BoolElt
    Cent : BoolElt : true
    Constructive : BoolElt : true
    Method : RngIntElt : 0
```

Given class ≤ 2, exponent p, indecomposable p-groups G and H of genus ≤ 2, decides if $G \cong H$. If $G \cong H$, then an isomorphism is provided unless specified otherwise. Currently, this does not work for $p = 2$. When the groups are decomposable, the intrinsic might decide isomorphism (e.g. when the centers properly contain the derived group), or it might return an error. In particular, an incorrect decision should not occur. There are three optional parameters: `Cent`, `Constructive`, and `Method`.

`Cent`: This parameter is a flag for the algorithm to rewrite the tensors over their centroids. In order to rewrite over the centroid, we assume the centroid is a local ring, see [**MW2**] for details. The default is `true`, so by default, the algorithm computes the centroid and rewrites the tensor over the residue field. If this step should be avoided, set to `false` to save some time.

`Constructive`: The default is `true`. Set to `false` if an explicit isomorphism is not needed.

`Method`: The default is 0, and any input from $\{0, 1, 2\}$ is acceptable. This will set the method for handling the sloped part of the tensor. If the adjoint-tensor method should be used, set to 1. If the Pfaffian method should be used, set to 2. The default will try to find the optimal method based on the input; this is not yet optimally tuned but is generally good.

```
TGIsPseudoIsometric(s, t : parameters) : TenSpcElt, TenSpcElt -> BoolElt
    Cent : BoolElt : true
    Constructive : BoolElt : true
    Method : RngIntElt : 0
```

Given indecomposable, alternating tensors $s, t : V \times V \rightarrowtail W$, where V and W are $\mathbb{F}_q$-vector spaces, decide if s is pseudo-isometric to t. If so, a pseudo-isometry is also returned as an element of $\mathrm{GL}(V) \times \mathrm{GL}(W)$ unless specified otherwise. Currently this does not work for even q. When the tensor are decomposable, the intrinsic might decide isomorphism (e.g. when the tensors have nontrivial radical), or it might return an error. In particular, an incorrect decision should not occur. There are three optional parameters: `Cent`, `Constructive`, and `Method`.

`Cent`: This parameter is a flag for the algorithm to rewrite the tensors over their centroids. In order to rewrite over the centroid, we assume the centroid is a local ring, see [**MW2**] for details. The default is `true`, so by default, the algorithm computes the centroid and rewrites the tensor over the residue field. If this step should be avoided, set to `false` to save some time.

`Constructive`: The default is `true`. Set to `false` if an explicit isomorphism is not needed.

`Method`: The default is 0, and any input from $\{0, 1, 2\}$ is acceptable. This will set the method for handling the sloped part of the tensor. If the adjoint-tensor method should be used, set to 1. If the Pfaffian method should be used, set to 2. The default will try to find the optimal method based on the input; this is not yet optimally tuned but is generally good.

CHAPTER 4

Automorphism groups

Like with isomorphisms, we have two distinct automorphism intrinsics.

```
TGAutomorphismGroup(G : parameters) : GrpPC -> GrpAuto
    Cent : BoolElt : true
    Method : RngIntElt : 0
    Mat : BoolElt : false
```

Given an indecomposable p-group G, of class ≤ 2, exponent p, and genus ≤ 2, return $\mathrm{Aut}(G)$. This currently does not work with $p = 2$. When G is decomposable, the intrinsic might construct $\mathrm{Aut}(G)$ (e.g. when the center properly contains the derived group), or it might return an error. In particular, the wrong (or incomplete) automorphism group should not be returned. This intrinsic provides three optional parameters: `Cent`, `Method`, and `Mat`.

`Cent`: This parameter is a flag for the algorithm to rewrite the tensors over their centroids. In order to rewrite over the centroid, we assume the centroid is a local ring, see [**MW2**] for details. The default is `true`, so by default, the algorithm computes the centroid and rewrites the tensor over the residue field. If this step should be avoided, set to `false` to save some time.

`Method`: The default is 0, and any input from $\{0, 1, 2\}$ is acceptable. This will set the method for handling the sloped part of the tensor. If the adjoint-tensor method should be used, set to 1. If the Pfaffian method should be used, set to 2. The default will try to find the optimal method based on the input; this is not yet optimally tuned but is generally good.

`Mat`: The default is `false`. When `true`, the output will be a linear representation of the automorphism group. The row vectors of the matrix correspond to the polycyclic vectors from the polycyclic presentation.

```
TGPseudoIsometryGroup(t : parameters) : TenSpcElt -> GrpMat
    Cent : BoolElt : true
    Method : RngIntElt : 0
```

Given an indecomposable, alternating tensor $t : V \times V \rightarrowtail W$, where V and W are $\mathbb{F}_q$-vector spaces, return the pseudo-isometry group $\Psi\mathrm{Isom}(t) \leq \mathrm{GL}(V) \times \mathrm{GL}(W)$. This currently does not work for even q. When t is decomposable, the intrinsic might construct $\Psi\mathrm{Isom}(t)$ (e.g. when t has a nontrivial radical), or it might return an error. In particular, the wrong (or incomplete) pseudo-isometry group should not be returned. This intrinsic provides two optional parameters: `Cent` and `Method`.

`Cent`: This parameter is a flag for the algorithm to rewrite the tensors over their centroids. In order to rewrite over the centroid, we assume the centroid is a local ring, see [**MW2**] for details. The default is `true`, so by default, the algorithm computes the centroid and rewrites the tensor over the residue field. If this step should be avoided, set to `false` to save some time.

`Method`: The default is 0, and any input from $\{0, 1, 2\}$ is acceptable. This will set the method for handling the sloped part of the tensor. If the adjoint-tensor method should be used, set to 1. If the Pfaffian method should be used, set to 2. The default will try to find the optimal method based on the input; this is not yet optimally tuned but is generally good.

CHAPTER 5

Canonical Labeling

`Genus2Signature(G) : GrpPC -> List`
`Genus2Signature(t) : TenSpcElt -> List`

Given a genus 2 indecomposable group or tensor, returns the canonical genus 2 signature as a list. The list has four parts. The first entry is a sequence of odd integers corresponding to the dimensions of the flat indecomposable spaces. The second entry is a list of sequences in $\mathbb{F}_q$ of possibly different lengths. Here, $\mathbb{F}_q$ is the residue field of the centroid $\mathcal{C}(\circ)$, which is a local ring. A sequence $[c_1, \ldots, c_{d+1}]$ in the second entry corresponds to the coefficients of the homogeneous polynomial in x and y in degree d:

$$c_1 x^d + c_2 x^{d-1} y + c_3 x^{d-2} y^2 + \cdots + c_{d+1} y^d.$$

Because this is a canonical label, two groups (or tensors) are isoclinic (or pseudo-isometric) if, and only if, their genus 2 signatures are equal.

When the input is decomposable, the intrinsic might construct the signature (e.g. when the tensor has a nontrivial radical), or it might return an error. In particular, the wrong signature should not be returned.

Bibliography

[BCP] 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. Computational algebra and number theory (London, 1993). MR1484478

[BMW] Peter A. Brooksbank, Joshua Maglione, and James B. Wilson, *A fast isomorphism test for groups whose Lie algebra has genus 2*, J. Algebra **473** (2017), 545–590. MR3591162

[BW] Peter A. Brooksbank and James B. Wilson, *StarAlge*, GitHub, 2019. https://github.com/thetensor-space/StarAlge.

[LGM] C. R. Leedham-Green and S. McKay, *The structure of groups of prime power order*, London Mathematical Society Monographs. New Series, vol. 27, Oxford University Press, Oxford, 2002. Oxford Science Publications. MR1918951

[MW1] Joshua Maglione and James B. Wilson, *Sylver*, GitHub, 2019. https://github.com/thetensor-space/Sylver.

[MW2] ———, *TensorSpace*, GitHub, 2019. Contributions from Peter A. Brooksbank, https://github.com/thetensor-space/TensorSpace.

Intrinsics