# Majorana and axial representations in GAP

Madeleine Whybrow

November 20, 2018

## 1 Getting started

To get the latest version of the `MajoranaAlgebras` package, go to `https://mwhybrow92.github.io/MajoranaAlgebras/` and download `MajoranaAlgebras-1.1.tar.gz`. Inside the `pkg` directory of your GAP installation, unpack `MajoranaAlgebras-1.1.tar.gz` by, for example, doing

```
tar -xzf MajoranaAlgebras-1.1.tar.gz
```

Start GAP in the normal way and call

```
LoadPackage("MajoranaAlgebras");
```

If this doesn't work, you may also need to install the following packages:

- automata;
- datastructures;
- Gauss.

## 2 Structures in GAP

*This section serves as an introduction to certain important GAP structures used in the Majorana algebras algorithm. Depending on your confidence level, feel free to spend more or less time on either of these topics, or to skip them completely.*

### 2.1 Sparse Matrices

Packages required: Gauss.
Manual reference: Gauss manual.

Matrices in GAP are considered as lists of lists. The GAP package Gauss offers an alternative way of storing and calculating matrices via the sparse matrix data type. At almost every point of the Majorana algorithm, computing with sparse matrices makes the calculations faster and the output clearer and so we store all matrices and vectors in this form.

1. Use `RandomMat` (GAP manual 24.6) to create a random matrix over `Rationals` as a list of lists.
2. Convert this matrix into a sparse matrix over `Rationals`. Hint: Gauss manual 3.2.1.

3. Calculate the rank of the matrix.

4. Calculate the echelonized form of this matrix. Hint: Gauss manual 4.2.1.

5. Print only the final row of the echelonized form of this matrix. Hint: Gauss manual 3.2.10.

### 2.2 Records

Packages required: none.
Manual reference: GAP manual Chapter 29

1. Create a record called `summer_school` such that

   - the components of `summer_school` are the titles of each of the six courses given here this week;
   - the value of each component is a record whose components are `speaker` and `time` and whose values are strings giving this information, e.g. `"Madeleine Whybrow"` and `"Tuesday pm"`.

   Hint: if the components of your record have names containing spaces, you might need to use brackets and strings to access their values (see 29.2 in the GAP manual).

2. The GAP code below constructs a list `tbl` of length two. Create a record `evecs` whose component names are the strings given in `tbl[1]` such that the value of the component `tbl[1][i]` is `tbl[2][i]`.

```
gap> tbl := [ [ "1", "0", "1/4", "1/32" ], [ ] ];;
gap> tbl[2][1] := SparseMatrix( 1, 8, [ [ 1 ] ], [ [ 1 ] ], Rationals);;
gap> tbl[2][2] := SparseMatrix( 3, 8, [ [ 2, 3, 4, 8 ], [ 1, 4, 7 ],
        [ 1, 2, 3, 4, 5, 6 ] ], [ [ -32/9, -32/9, -8/9, 1 ],
        [ -1/4, 1, 1 ], [-5/16, 3, 3, 3/4, 1, 1 ] ], Rationals );;
gap> tbl[2][3] := SparseMatrix( 2, 8, [ [ 1, 5, 6, 8 ], [ 4, 7 ] ],
        [ [ -8/45, -32/45, -32/45, 1 ], [ -1, 1 ] ], Rationals );;
```

3. Write a function that performs the procedure described in the previous question for a generic list `tbl` of length two such that `Length(tbl[1]) = Length(tbl[2])`. Hint: GAP tutorial Chapter 4.

## 3 Majorana algebras

*This section guides you through the construction of a Majorana representation of your choice and suggests some calculations that you can perform on it, once you have constructed it.*

### 3.1 Construct a Majorana representation

1. Choose a group $G$ and a subset $T$ of involutions of $G$ such that $(G, T)$ is a 6-transposition group.

   Suggestions of such groups include:

- the symmetric and alternating groups;
- elementary abelian groups of order $2^n$;
- 3-transposition groups.

*Note that the algorithm to construct Majorana representations is expensive both in terms of time and memory - you will have more success if you choose $G$ to be small (around 500 elements or fewer).*

For example:

```
gap> G := SU(3,2);;
gap> cc := ConjugacyClasses(G);;
gap> cc := Filtered(cc, x -> Order(Representative(x)) = 2);;
gap> Size(cc);
1
```

Here there is only one conjugacy class of involutions of $G$ so we have only one choice for the set $T$ - this might not be the case in your group.

```
gap> T := AsList(cc[1]);;
gap> Size(G);
216
gap> Size(Group(T));
54
```

The group that we have chosen is not a 6-transposition group as it is not generated by $T$. This is not a problem - we can instead take $G$ to be the subgroup generated by $T$.

```
gap> G := Group(T);;
gap> StructureDescription(G);
"((C3 x C3) : C3) : C2"
gap> MAJORANA_IsSixTranspositionGroup(G,T);
true
```

2. For your chosen $G$ and $T$, calculate the possible shapes of a Majorana representation of the form $(G, T, V)$.

```
gap> input := ShapesOfMajoranaRepresentation(G,T);;
gap> Size(input.shapes);
16
```

Here there are a lot of shapes so we use the function `MAJORANA_RemoveDuplicateShapes` to see if any can be removed.

```
gap> MAJORANA_RemoveDuplicateShapes(input);
gap> Size(input.shapes);
5
gap> input.shapes;
[ [ "1A", "3C", "3C", "3C", "3C" ], [ "1A", "3C", "3C", "3C", "3A" ],
  [ "1A", "3C", "3C", "3A", "3A" ], [ "1A", "3C", "3A", "3A", "3A" ],
  [ "1A", "3A", "3A", "3A", "3A" ] ]
```

3. For one or more of the shapes given by $G$ and $T$, (attempt to) construct the Majorana representation with this shape.

```
gap> rep := MajoranaRepresentation(input, 5);;
gap> reps := List( [1 .. 5], i -> MajoranaRepresentation(input, i));;
```

If the function is taking a long time to run, first call `SetInfoLevel(InfoMajorana, 100)`. The function will then output some information that might persuade you that it is progressing. If this is not the case, then you might want to try again with a smaller group or a different shape.

```
gap> SetInfoLevel(InfoMajorana, 100);
gap> rep := MajoranaRepresentation(input, 5);;
#I  Axiom M1
#I  Fusion of 1 evecs
#I  Building eigenvector unknowns
#I  Building nullspace unknowns
#I  Building resurrection
#I  Solved a single solution
#I  Solved a single solution
#I  Solved a single solution
#I  Solved a single solution
#I  Solved a single solution
#I  Solved a single solution
#I  There are 0 unknown algebra products
#I  There are 0 unknown inner products
#I  Success
```

Once the function has finished, call `MAJORANA_IsComplete` on the output. If this returns `true` then you are done and can move on to the next section.

```
gap> MAJORANA_IsComplete(rep);
true
```

Otherwise, you can then call the function `NClosedMajoranaRepresentation` on the output to attempt construction of the 3-closed part of the algebra. If this is again taking too long then you should try a different group or shape.

## 3.2 Calculating with Majorana algebras

Now that you have constructed a (complete) Majorana representation using the previous section, you can start calculations on it. Questions 1 - 4 should be straightforward with the help of the package manual and Section 2 of this tutorial. Questions 5 - 8 require a little more thought.

1. What is the dimension of the representation?
2. What is the size of the spanning set `coords` of the representation?
3. What is the dimension of the nullspace of the representation?
4. What are the dimensions of the eigenspaces of a given Majorana axis?
5. Which of the spanning set vectors are idempotents?
6. If one exists, pick an idempotent that is not a Majorana axis. What are the eigenvalues of its adjoint action on the representation?
7. What is the fusion law of the eigenspaces of this idempotent?
8. Are there any other idempotents that also have this fusion law? If so, what is the dimension of the subalgebra that they generate?

If you get as far as questions 7 and 8, I am interested to know what answers you come up with!