Majorana and axial representations in GAP

Madeleine Whybrow

November 20, 2018

1 Getting started

2 Structures in GAP

2.1 Sparse Matrices

1. Use `RandomMat` (GAP manual 24.6) to create a random matrix over `Rationals` as a list of lists.

```
gap> mat := RandomMat(20, 30, Rationals);;
```

2. Convert this matrix into a sparse matrix over `Rationals`. Hint: Gauss manual 3.2.1.

```
gap> sparse_mat := SparseMatrix(mat, Rationals);;
```

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

```
gap> ech := EchelonMat(sparse_mat);;
```

4. Calculate the rank of the matrix.

```
gap> rank := Rank(sparse_mat);
gap> # Or
gap> rank := Nrows(ech.vectors);
```

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

```
gap> last_row := CertainRows( ech.vectors, [ rank ] );;
gap> Print( last_row );
```

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"`;

```
gap> summer_school := rec();;
gap> summer_school.("GAP tutorial") := rec(
                                    speaker := "Alexander Konovalov",
                                    time := "Monday");;
gap> summer_school.("The CAP Project") := rec(
                                    speaker := "Sebastian Posur",
                                    time := "Tuesday am");;
gap> # etc.
```

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 );;
gap> tbl[2][4] := SparseMatrix( 2, 8, [ [ 5, 6 ], [ 2, 3 ] ],
        [ [ -1, 1 ], [ -1, 1 ] ], Rationals );;
gap> record := rec();;
gap> for i in [ 1 .. Size(tbl[1]) ] do
        record.(tbl[1][i]) := tbl[2][i];
    od;
```

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.

```
gap> ConvertTableToRecord := function(tbl)
        local i, record;

        record := rec();

        for i in [1 .. Size(tbl[1])] do
            record.( String( tbl[1][i] ) ) := tbl[2][i];
        od;

        return record;

    end;
```

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

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?

```
gap> MAJORANA_Dimension(rep);
```

2. What is the size of the spanning set `coords` of the representation?

```
gap> Size(rep.setup.coords);
```

3. What is the dimension of the nullspace of the representation?

```
gap> Nrows(rep.setup.nullspace.vectors);
```

4. What are the dimensions of the eigenspaces of a given Majorana axis?

```
gap> for ev in rep.eigenvalues do
        Display( Nrows( MAJORANA_Eigenvectors( 1, ev, rep) ) );
    od;
```

5. Which of the spanning set vectors are idempotents?

```
gap> s := Size(rep.setup.coords);;
gap> idempotents := [];;
gap> for i in [1..s] do
        u := SparseMatrix( 1, s, [[i]], [[1]], Rationals);
        u2 := MAJORANA_AlgebraProduct(u, u, rep.algebraproducts, rep.setup);
        if u = u2 then
            Add(idempotents, i);
        fi;
    od;
```

6. If one exists, pick an idempotent that is not a Majorana axis. What are the eigenvalues of its adjoint action on the representation?

```
gap> # Find an idempotent that is not a Majorana axis
gap> i := First(idempotents, i -> i > Size(rep.involutions) );
gap> # Make this into a row vector
gap> axis := SparseMatrix(1, s, [[i]], [[1]], Rationals);;
gap> # Find a basis of the algebra
gap> basis := MAJORANA_Basis(rep);;
gap> # Calculate the adjoint action of <axis> with respect to this basis
gap> adj := MAJORANA_AdjointAction(axis, basis, rep);;
gap> # Then you need to convert this to a list of lists matrix
gap> adj_mat := ConvertSparseMatrixToMatrix(adj);;
gap> # And use this to find the eigenvalues
gap> Eigenvalues( Rationals, adj_mat );
[1, 1/3, 0]
```

7. What is the fusion law of the eigenspaces of this idempotent?

```
gap> # Construct each of the eigenspaces
gap> zero := KernelMat(adj).relations;;
gap> one := KernelMat(adj - SparseIdentityMatrix(13, Rationals)).relations;;
gap> third := KernelMat(adj - (1/3)*SparseIdentityMatrix(13, Rationals)).relations;;
gap> # Construct a basis of eigenvectors
gap> espace := UnionOfRows(zero, one);;
gao> espace := UnionOfRows(espace, third);;
gap> # Find the products of, for example, zero eigenvectors with zero eigenvectors
gap> zero_zero := SparseMatrix(0, 13, [], [], Rationals);;
gap> for i in [1..Nrows(zero)] do
>       v := CertainRows(zero, [i]);
>       for j in [1..Nrows(zero)] do
>               u := CertainRows(zero, [j]);
>               prod := MAJORANA_AlgebraProduct(u, v, rep.algebraproducts,rep.setup);
>               zero_zero := UnionOfRows(zero_zero, prod);;
>       od;
```

```
> od;
gap> zero_zero := EchelonMatDestructive(zero_zero).vectors;;
gap> # Now find where these vectors lie in our eigenspace decomposition
gap> x := SparseMatrix(0, 13, [], [], Rationals);;
gap> for i in [1..6] do
>       v := CertainRows(zero_zero, [i]);;
>       x := UnionOfRows(MAJORANA_ConvertToBasis(espace, v), x);;
> od;
gap> x!.indices;
[ [ 6 ], [ 5 ], [ 4 ], [ 2, 3 ], [ 3 ], [ 1, 3 ] ]
```

So these vectors all lie in the first six rows of the eigenspace, so are all zero eigenvectors. The fusion law of these idempotents ends up being

$*$	1	0	$\frac{1}{3}$
1	1	$\emptyset$	$\frac{1}{3}$
0	$\emptyset$	0	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$1, 0, \frac{1}{3}$

8. Are there any other idempotents that also have this fusion law? If so, what is the dimension of the subalgebra that they generate?

```
gap> vecs := SparseMatrix(4, 13, [[10], [11], [12], [13]],
>                               [[1], [1], [1], [1]], Rationals);;
gap> subalg := MAJORANA_Subalgebra(vecs, rep);;
gap> Nrows(subalg);
4
```

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