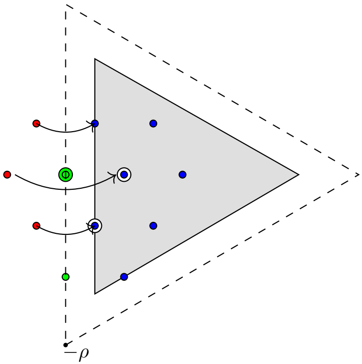
FusionRings in Sage 9.2 and beyond

Daniel Bump



Verlinde Algebras

The `FusionRing` method in Sage has methods for working with Verlinde algebras, which are the Grothendieck rings of certain Modular Tensor Categories (MTC) that arise in different contexts.

- Wess-Zumino-Witten conformal field theories
- Representations of quantum groups at roots of unity
- Representations of affine Lie algebras

These MTC and Verlinde algebras have some “magical” properties and applications

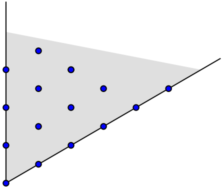
- Knot invariants and braid group representations
- Applications in topological quantum computing

A similar familiar category

Let G be a compact Lie group. The irreducible representations are finite-dimensional and are in bijection with the dominant weights. These are lattice points in a cone, the positive Weyl chamber

$$\{v \in X^*(T) | \langle v, \alpha_i^\vee \rangle \geqslant 0\},$$

$T =$ maximal torus, $\alpha_i^\vee =$ simple coroots.



example: $G = SU(3)$

The WeylCharacterRing

The Grothendieck group of the category of finite dimensional representations of G is the `WeylCharacterRing`. It has a basis parametrized by the dominant weights.

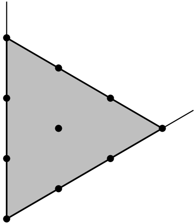
The multiplication decomposes tensor products into irreducibles.

```
sage: A2=WeylCharacterRing("A2",style="coroots")
sage: A2(1,1)*A2(4,2)
A2(3,1) + A2(2,3) + A2(5,0) + 2*A2(4,2) + A2(3,4)
  + A2(6,1) + A2(5,3)
```

It has further methods to compute weight multiplicities, symmetric and exterior powers, Frobenius-Schur indicators, and branching rules.

The Fusion Category of Level k

The Fusion Category of the Level k Wess-Zumino-Witten conformal field theory has a similar structure to the Weyl character ring. However we truncate the Weyl chamber at level k resulting in the level k fundamental alcove:



$$\langle x, \alpha_i^\vee \rangle \geqslant 0,$$

$$\langle x, \alpha_{\text{highest}}^\vee \rangle \leqslant k.$$

In this example $G = SU(3)$ and $k = 3$. There are 10 simple objects in this monoidal category.

Contributors

The `FusionRing` code in Sage was mostly written by Daniel Bump and Guillermo (Willie) Aboumrad, with support from Travis Scrimshaw. This code is already merged in Sage, and it can do *most* of the calculations for applications.

The F-matrix code, which is still under development, is needed to complete the picture. With this, working F-matrix code was prototyped by Galit Anikeeva, Daniel Bump and Guillermo Aboumrad in:

Trac Ticket #30423 (web link).

Recently major improvements have been found by Aboumrad, allowing much larger cases and computations of braid group representations. We hope these will be available soon.

Anyons

One application of Fusion categories is to computing the properties of nonabelian anyons, which are quasiparticles that emerge in connection with the quantum Hall effect. Potentially these can be harnessed for quantum computing.

Anyons are neither bosons nor fermions, for interchanging two particles produces a phase shift in the wave function that can be a root of unity, not necessarily ± 1. Such particles can only exist is two dimensional systems. In the quantum Hall effect the particles are constrained to a two-dimensional plate.

Permuting the particles produces an entangled state. To compute it, one must work with a braid group representation. We will explain how the fusion code can compute these.

Some references

- Bakalov and Kirillov, Lectures on tensor categories and modular functors, AMS (2001).
- Rowell and Wang, Mathematics of topological quantum computing. Bull. AMS 55 (2018). arXiv:1705.06206
- Di Francesco, Mathieu and Senechal, *Conformal Field Theory*, Springer 1997, Chapter 16
- Turaev, *Quantum invariants of knots and 3-manifolds*, de Gruyter 1994, 2010 and 2016
- P. Bonderson, Nonabelian anyons and interferometry, Dissertation (2007). https://thesis.library.caltech.edu/2447/
- Z. Wang, Topological quantum computation. Providence, RI: American Mathematical Society (AMS), 2010.

The FusionRing

The Grothendieck group of the level k fusion category may be constructed from the `FusionRing` method. This class is very similar to the `WeylCharacterRing` and in fact inherits from it as a Python class. You can name the simple objects whatever you want.

```
sage: I = FusionRing("E8",2,conjugate=True)
sage: I.fusion_labels(["i0","p","s"],inject_variables=True)
sage: b = I.basis().list(); b
[i0, p, s]
sage: [[x*y for x in b] for y in b]
[[i0, p, s], [p, i0, s], [s, s, i0 + p]]
```

However the `FusionRing` has a number of quantum methods that require explanation.

```
sage: [(b,b.\rred{ribbon}()) for b in I.basis()]
[(i0, 1), (p, -1), (s, zeta128^8)]
```

The FusionRing (continued)

The code in Sage 9.2 is complete except for the F-matrix code, and includes methods for the quantum dimensions, S-matrix, twists, Virasoro central charge, etc. As currently implemented, methods output elements of a cyclotomic field. An option for `QQbar` output may be added.

Sage 9.3 will have a bugfix in the R-matrix (#30423.)

The F-matrix code will be available in either Sage 9.3 or 9.4. An early version is in a git branch linked from #30423. A better version should be available soon.

- Reference Manual Page for `FusionRing`
- Trac Ticket #30423

Monoidal Categories

The quantum methods of the `FusionRing` reflect the fact that the fusion category is ribbon.

A ribbon category is first of all a monoidal category, meaning that it has an “tensor” bifunctor $\otimes$ that is associative in that there is given a natural associator isomorphism

$$\alpha : (A \otimes B) \otimes C \longrightarrow A \otimes (B \otimes C)$$

such that the Pentagon Relation is satisfied:

$$\begin{tikzcd}
 & (A \otimes B) \otimes (C \otimes D) \arrow[dr] & \\
((A \otimes B) \otimes C) \otimes D \arrow[ur] \arrow[d] & & A \otimes (B \otimes (C \otimes D)) \\
(A \otimes (B \otimes C)) \otimes D \arrow[r] & A \otimes ((B \otimes C) \otimes D) \arrow[ur] &
\end{tikzcd}$$

Rigid Braided Categories

A monoidal category is braided if it also has a natural commutativity morphism

$$c_{A,B} : A \otimes B \to B \otimes A$$

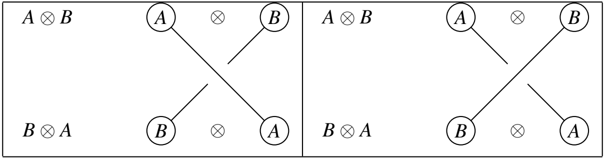
subject to the hexagon relations:

$$\begin{tikzcd}
 & A \otimes (B \otimes C) \arrow[r] & (B \otimes C) \otimes A \arrow[dr] & \\
(A \otimes B) \otimes C \arrow[ur] \arrow[dr] & & & B \otimes (C \otimes A) \\
 & (B \otimes A) \otimes C \arrow[r] & B \otimes (A \otimes C) \arrow[ur] &
\end{tikzcd}$$

We also assume the mirror image relation (not shown).

Graphical representation

The morphisms $c_{A,B} : A \otimes B \to B \otimes A$ and $c_{B,A} : B \otimes A \to A \otimes B$ are not inverses. So $c_{A,B}$ and $c_{B,A}^{-1}$ are distinct morphisms $A \otimes B \to B \otimes A$. We represent these graphically:

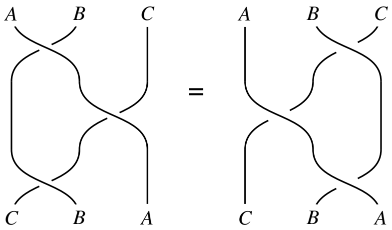


Read these from top to bottom.

The Yang-Baxter equation

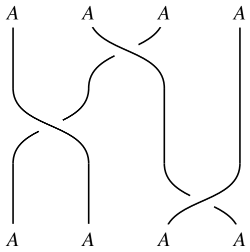
The hexagon equations plus naturality imply commutativity of:

$$\begin{tikzcd}
 & B \otimes A \otimes C \arrow[r] & B \otimes C \otimes A \arrow[dr] & \\
A \otimes B \otimes C \arrow[ur] \arrow[dr] & & & C \otimes B \otimes A \\
 & A \otimes C \otimes B \arrow[r] & C \otimes A \otimes B \arrow[ur] &
\end{tikzcd}$$



The braid group action

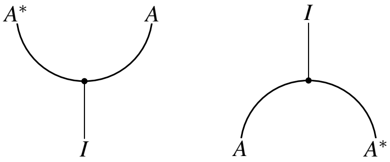
Given an object in a braided category, using the Yang-Baxter equation, we obtain an action of the Artin braid group $\mathfrak{B}_n$ (with n strands) on $\bigotimes^n A$ for any object A:



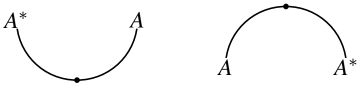
Fusion categories are braided, so they come with braid group actions, and these braid group actions are important in topological quantum computing.

Rigid categories

A monoidal category has an identity element I such that $A \otimes I \cong I \otimes A \cong A$. In a rigid category every object A has a left dual A^* with morphisms $\mathrm{ev} : A^* \otimes A \to I$ and $\mathrm{coev} : I \to A \otimes A^*$.



We suppress the I:

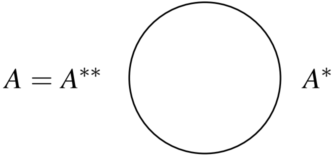


Quantum Dimension

In a rigid braided category if $A \cong A^{**}$ we may compose the evaluation and coevaluation:

$$I \xrightarrow{\quad \text{coev} \quad} A \otimes A^* \xrightarrow{\quad \text{ev} \quad} I$$

This is a morphism $I \to I$, thus a scalar, the quantum dimension.

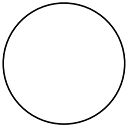


It is only defined if $A = A^{**}$ and it only has nice properties if the category is ribbon. In a braided rigid category the isomorphism $A \cong A^{**}$ is true but only in a ribbon category is there a distinguished isomorphism that makes everything work.

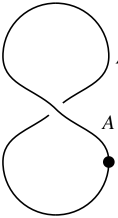
Ribbon Categories

In a ribbon category every element has A an endomorphism θ_A called the twist or ribbon element with magical properties. The isomorphism $I \to A \otimes A^* \to I$ becomes more precisely the composition

$$I \xrightarrow{\text{coev}} A \otimes A^* \xrightarrow{c_{A,A^*}} A^* \otimes A \xrightarrow{1 \otimes \theta_A^{-1}} A^* \otimes A \xrightarrow{\text{ev}} I$$



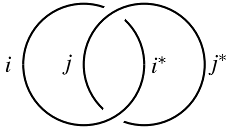
becomes



Modular Tensor Categories

We have already seen that level k WZW fusion category has only finitely many simple objects. It is a ribbon category, so it has a ribbon element.

If i, j are simple objects consider the Hopf link:



This is an endomorphism $I \to I$, that is a scalar $s_{i,j}$. The matrix $S = (s_{i,j})$ is called the S-matrix. Many things can be computed in terms of it.

Fusion Ring Methods in Sage

Given a modular tensor category, the following quantities are needed:

- Fusion Rules (multiplication in the `FusionRing`)
- Quantum dimensions of simple objects
- Twists
- Total quantum order
- Central charge
- S-matrix

Sage can compute all of the above. Two more quantities require discussion:

- R-matrix (yes with caveats)
- F-matrix (coming soon: trac ticket #30423)

Example: the Fibonacci MTC

Here are some computations for a well-known example, the Fibonacci MTC.

```
sage: F = FusionRing("G2",1) # Fibonacci anyons
sage: F.fusion_labels(["i0","t"],inject_variables=True)
sage: [i0*i0,i0*t,t*t] # fusion rules
sage: [x.q_dimension() for x in F.basis()]
sage: phi = t.q_dimension()
sage: phi^2==phi+1
True
sage: F.virasoro_central_charge()
14/5
sage: F.s_matrix()
[                              1 -zeta60^14 + zeta60^6 + zeta60^4]
[-zeta60^14 + zeta60^6 + zeta60^4                             -1]
```

The S-matrix can be alternatively normalized to be unitary, and the individual entries can also be had.

Multiplicity-free case

An important special case is when the Fusion category is multiplicity free meaning that if i, j, k are irreducible then the multiplicity N_k^{ij} of i in $i \otimes j$ is $\leqslant 1$.

The multiplicity-free case is an important one. For example, the MTC that arise in connection with topological quantum computing schemes are usually multiplicity-free.

Two things may be more easily studied in the multiplicity-free case.

- The R-matrix or 3j-symbol
- The F-matrix or 6j-symbol

The R-matrix

The R-matrix is a synonym for the commutativity morphism $c_{A,B} : i \otimes j \to j \otimes i$ in our previous notation. The module $i \otimes j$ may be reducible, so this is not a simple scalar.

We assume that $i \otimes j$ is multiplicity-free for all simple i, j. If k occurs in $i \otimes j$, let $\gamma_k^{i,j} : k \to i \otimes j$ be a fixed injection. It is determined up to scalar. The data $\{\gamma_k^{i,j}\}$ are called a gauge.

Once a gauge is fixed, the R-matrix has tangible meaning.

$$\begin{tikzcd}
k \arrow[r, "R_k^{i,j}"] \arrow[d, "\gamma_k^{i,j}"] & k \arrow[d, "\gamma_k^{j,i}"] \\
i \otimes j \arrow[r, "c_{i,j}"] & j \otimes i
\end{tikzcd}$$

The top arrow is a scalar that Sage can compute.

Definition of the F-matrix

Just as the R-matrix captures the commutativity morphism, the F-matrix captures the associator

$$\alpha_{i,j,k} : (i \otimes j) \otimes k \to i \otimes (j \otimes k).$$

Let l be a another object. Then $\alpha_{i,j,k}$ induces a linear map

$$\mathrm{Hom}(l, (i \otimes j) \otimes k) \to \mathrm{Hom}(l, i \otimes (j \otimes k)).$$

We have bases of these Hom spaces as follows.

Definition of the F-matrix (continued)

Let x and y be auxiliary objects. A basis for $\mathrm{Hom}(l, (i \otimes j) \otimes k)$ consists of the composite morphisms

$$l \xrightarrow{\gamma_l^{x,k}} x \otimes k \xrightarrow{\gamma_x^{i,j} \otimes 1_k} (i \otimes j) \otimes k \quad .$$

Call this composition $\phi_{i,j,k}^x$. Let $\psi_{i,j,k}^y$ be the morphism $\mathrm{Hom}(l, i \otimes (j \otimes k))$ that is the composition

$$l \xrightarrow{\gamma_l^{i,y}} i \otimes y \xrightarrow{1_i \otimes \gamma_y^{j,k}} i \otimes (j \otimes k) \quad .$$

The F-matrix is the matrix of the linear transformation $\mathrm{Hom}(l, (i \otimes j) \otimes k) \to \mathrm{Hom}(l, i \otimes (j \otimes k))$ induced by $\alpha_{i,j,k}$ in terms of the bases $\phi_{i,j,k}^x$ and $\psi_{i,j,k}^y$ of these two Hom spaces.

Computing the R-matrix

The R-matrix $R_k^{i,j}$ may be expressed in terms of the S-matrix by a formula of Bonderson, Delaney, Galindo, Rowell, Tran and Wang. The difficult case is when $i=j$. In that case, these authors proved

$$R_k^{i,i} = \sum_{x,y,z} \frac{\theta_y^2}{\theta_i \theta_x^2} \frac{S_{0,y} S_{i,z} \overline{S_{x,z} S_{k,x} S_{y,z}}}{S_{0,z}}$$

where θ_x is the ribbon element (a root of unity) and $S_{x,y}$ is the S-matrix, normalized to be unitary.

Computing the F-matrix

By contrast, computing the F-matrix is a delicate matter. Unlike all the other methods we have encountered, F-matrix is not given by a simple formula in terms of the S-matrix, as far as we know.

Bonderson's thesis describes a nuanced approach to computing the F-matrices. Last summer, Galit Anikeeva, Willie Aboumrad and I implemented an "F-matrix factory" in Trac Ticket #30423. This was able to handle small cases.

Recently Aboumrad has much better code that can compute much larger cases. It is hoped that this will be available soon.

The pentagon equations

The pentagon equations:

$$
\begin{tikzcd}
 & (A\otimes B)\otimes(C\otimes D) \arrow[dr] & \\
((A\otimes B)\otimes C)\otimes D \arrow[ur] \arrow[d] & & A\otimes(B\otimes(C\otimes D)) \\
(A\otimes(B\otimes C))\otimes D \arrow[r] & A\otimes((B\otimes C)\otimes D) \arrow[ur] &
\end{tikzcd}
$$

impose conditions on the values of the F-matrix. These are algebraic equations, and in small cases we can try to solve them using Groebner basis methods.

Pentagon equations only

There are many solutions to the pentagon equations, but these can be adjusted using the gauges, so effectively there is only one solution.

However the pentagon equations are rather diffuse and difficult to solve without making use of further information. We may make a graph on the variables, joining two variables in an edge if they occur in the same equation. The graph of the pentagon equations is very highly connected.

For B_2 at level 2, there are 725 unknowns in the F-matrix. These must satisfy 13175 equations coming from the pentagon relations. Solving these is a difficult task.

Hexagon equations

The hexagon equations also constrain the F-matrix.

$$\begin{tikzcd}
& A \otimes (B \otimes C) \arrow[r] & (B \otimes C) \otimes A \arrow[dr] & \\
(A \otimes B) \otimes C \arrow[ur] \arrow[dr] & & & B \otimes (C \otimes A) \\
& (B \otimes A) \otimes C \arrow[r] & B \otimes (A \otimes C) \arrow[ur] &
\end{tikzcd}$$

We can try to solve the pentagon relations without the hexagon equations, but actually the hexagon relations are very helpful. For the B_2 level 2 example, there are 725 hexagon equations including useful ones like:

```
fx715 + 1 == 0
fx563*fx707 + (-zeta40^8)*fx564*fx709
        + (-zeta40^14)*fx681 == 0
```

Why the Hexagon Equations

It is important to use the hexagon equations in addition to the pentagon equations for two reasons:

- The equation graph for the hexagon equations has more favorable localization properties, making the system easier to solve.
- Unlike the pentagon equations, the hexagon equations include information from the R-matrix. Having set the R-matrix already partially fixes the gauge. On solving the pentagon equations without the hexagon equations, the resulting solutions may not be consistent with the hexagon equations due to the gauge being set twice.
- The hexagon equations bring in roots of unity from the R-matrix making possible a cyclotomic solution.

Information from unitaricity

Bonderson pointed out in his thesis that the F-matrix can be made unitary and this gives information about which F-matrix values are automatically unitary. Aboumrad found in recent that by including the conditions of unitaricity with the pentagon and hexagon equations, much larger cases can be solved.

The unitary F-matrix is not cyclotomic, so unlike most of the `FusionRing` methods, if unitaricity is imposed the solutions will be in `QQbar`.

It is intended that these methods will be included in later versions of Sage.