A Mathematica package for quantum group representation theory.

SCOTT MORRISON

Email:

URL:

Abstract

AMS Classification ;

Keywords

Contents

Computing quantum knot invariants.

What are the Reshetikhin-Turaev invariants **[?]** of links coming from quantum groups? For each quantum group $U_q(\mathfrak{g})$ (by which we mean the quantised universal enveloping algebra of a complex simple Lie algebra $\mathfrak{g}$, see below), we have a function

$$(\text{framed links, with components labelled by irreps of } U_q(\mathfrak{g})) \to \mathbb{Z}[q, q^{-1}].$$

In this paper, I describe how one computes these invariants. In particular, I'll tell you just enough mathematics for the definition, but much more importantly, I'll tell you how to *actually* compute them, by showing you how to use a Mathematicapackage called `QuantumGroups‘`.

In fact, the package does much more than just compute quantum knot invariants. Subject to quite restrictive practical limitations,[1] `QuantumGroups‘` can

- Calculate dimensions of weight spaces and invariant spaces of tensor products of arbitrary highest weight representations, using a combinatorial model.

[1] My code is inefficient, the algorithms are slow, and the computations are difficult!

- Produce matrices representing the action of the generators of the quantum group $U_q(\mathfrak{g})$ on an arbitrary highest weight representation.
- Calculate bases for the invariants spaces inside tensor products of representations, or bases for intertwining maps between two such tensor products.
- Calculate the action of the universal R-matrix on pairs of representations.

By the end, you'll understand how to answer questions like:

> What is the invariant of the knot 8_{19} **TODO: picture!**, labelled by the 14 dimensional irrep of G_2?

(For the really impatient, one way is to download the `KnotTheory‘` Mathematica-package from `http://katlas.org/`, and enter[2] the following in Mathematica:

In[1]:=`<<KnotTheory‘`

> Loading KnotTheory‘ version of January 18, 2008, 18:17:28.7446.
> Read more at `http://katlas.org/wiki/KnotTheory`.

In[2]:=`QuantumKnotInvariant[G2,Irrep[G2][0,1]][Knot[8,19]][q]` **TODO: check this works!**

Out[2]=`???`

1 What's already done?

The Reshetikhin-Turaev invariants have been around for quite a while, but there hasn't been a significant tabulation of calculations, or a general purpose program to compute them. In this section I'll summarise what's already known. I'll concentrate on mentioning general purpose programs, which work for arbitrary links (or perhaps just knots). There's certainly more to say for many particular families of links.

The Jones polynomial [**?**] is the first interesting special case, when $\mathfrak{g} = 2$, and each component of the link is labelled with the two dimensional representation. Of course programs to compute this abound [**?**], as do tabulations of the invariants [**?**]. From the Jones polynomial, we can generalise in two directions:

(1) Labelling the link with other irreps of $U_q(\mathfrak{sl}_2)$. When all the labels are the $n+1$ dimensional irrep, this is called the n-th coloured Jones polynomial of the link.

(2) Using the quantum group $U_q(\mathfrak{sl}_n)$, and labelling each component by the standard n dimensional irrep.

Again, there are many programs available which calculate both of these invariants, and many tabulations. It's a little unusual to see direct discussion of the invariant coming from the standard representation of $U_q(\mathfrak{sl}_n)$, however, because it turns out that these invariants, for varying n, all fit together as a two variable polynomial, the HOMFLYPT polynomial [**?**]. In particular,

[2]Don't type 'In[1]:='; Mathematicawill add this itself. See §?? for more details.

$$HOMFLYPT_K(q^n, q) = RT_{U_q(\mathfrak{sl}_n),\mathbb{C}^n}(K)(q).$$

Thus to find programs or tables of these invariants, you're for the most part better off looking for the HOMFLYPT invariant. One notable exception is a program available in the `KnotTheory‘` Mathematicapackage [**?**], which makes a direct calculation of the $U_q\left(\mathfrak{sl}_3\right)$ invariant, via Kuperberg's spider [**?**].

Next, the two variable Kauffman polynomial simultaneously captures all the Reshetikhin-Turaev invariants for the standard representations of the quantum groups $U_q(\mathfrak{so}(n))$, $n \geq 5$, and $U_q(\mathfrak{sp}(n))$, $n \geq 4$.

TODO: look these up, and write some formulas

—obsolete from here on—

In this paper, we describe a Mathematicapackage for performing computations in the representation theory of an arbitrary quantum group $U_q\left(\mathfrak{g}\right)$. this package can

As an application, we demonstrate the computation of Reshetikhin-Turaev quantum knot invariants in many cases which have not been previously calculated.

This paper is instructional, rather than expository; we give examples of the usage of the package, but present no 'new mathematics'.

2 Installing the `QuantumGroups‘` package

3 Combinatorial representation theory

4 Explicit representations

The function `MatrixPresentation` produces explicit matrices representing the action of the quantum group generators on a representation.

It is invoked as `MatrixPresentation`$[\Gamma][Z][V, \lambda, \beta]$. Here

Definition 4.1

Γ *is the Cartan type, see* §**??**.

Z *is a generator of the quantum group* Γ*, that is* $X_i^{\pm}$ *or* K_i*, for* $1 \leq i \leq \mathrm{rank}\left(\Gamma\right)$*. Compositions of generators, in the notation of* **??***, and linear combinations, are also allowed. (Linear combinations must be homogeneous with respect to the weight grading.)*

V *is a representation, in the notation of* §**??**.

λ *is a weight, in the notation of* §**??***; that is, a vector of integers, giving the weight as a linear combination of fundamental weights.*

4.1 Bases

The function `MatrixPresentation` takes an argument specifying the desired basis. In the current implementation, there is only one useful value – the symbol `FundamentalBasis`. While we give a description of how this basis is recursively defined below, essentially it depends on many minor details of the implementation. One should not depend on any particular properties of this basis!

Future versions of the `QuantumGroups‘` package may allow the use of the symbols `GelfandTsetlinBasis` and `CanonicalBasis`, with the obvious results.[3] Code implementing Gelfand-Testlin bases exists, but is not currently part of the package. Anyone interested in adding support for canonical bases should certainly contact me!

5 Invariant vectors and intertwiners

6 R-matrices and quantum knot invariants

References

This paper is available online at arXiv:`?????`, and at `http://scott-morrison.org/quantum_groups`.

Email:
URL:

[3]Gelfand-Tsetlin bases are only projectively canonical.