THE JONES POLYNOMIAL AND KHOVANOV HOMOLOGY: IMPLEMENTATIONS IN *SNAPPY*, *KNOTKIT*, *SAGE MATH*

ERNEST YEUNG ERNESTYALUMNI@GMAIL.COM

Contents

Abstract. I share my implementation in *SnapPy*, `knotkit`, and *Sage Math* of the Jones polynomial $J(K,q)$ and the "categorified" Jones Polynomial $J(K;t,q)$, computed via Khovanov Homology, for torus knots K, and show that setting $t=-1$ for $J(K;t,q)$ reproduces $J(K,q)$. The directory `JonesPoly_and_KH` of github repository `ernestyalumni:qSApoly` contains text (string) files containing all polynomials, $J(K;q)$ and $J(K;t,q)$, for torus knots of less than 14 crossings-the purpose is to "standardize" and agree upon conventions for Jones polynomials and "categorified" Jones polynomials via Khovanov homology. I also point out avenues for further investigations.

1. Jones Polynomials from SnapPy

One should have installed successfully SnapPy and its implementation in Sage Math (cf. 3.1).

Then, one can obtain the Jones polynomial for various torus knots in such a manner - in Sage Math:

```
sage: import snappy
sage: T0203snap = snappy.Link('T(2,3)')
sage: T0203snap # obtain the number of crossings as a print out!
<Link: 1 comp; 3 cross>
sage: T0203snap.crossings # obtain the crossings as a list!
[0, 1, 2]
sage: len(T0203snap.crossings) # this is another way to obtain the number of crossings
3
```

If one does `dir(T0203)`, then as one can see, there are a number of modules that one can play with. For instance, the signature and morse number can be obtained quickly:

```
sage: T0203snap.signature()
2
sage: T0203snap.morse_number()
2
```

and also the Jones polynomial can be obtained quickly, after declaring the torus knot with `Link`:

```
sage: T0203snap.jones_poly()
 q^4 + q^3 + q
```

Date: 24 février 2016.
Key words and phrases. Jones Polynomial, Khovanov Homology, low-dimensional topology.

Let's obtain all the Jones polynomials for torus knots with less than 14 crossings (arbitrary choice of 14) with SnapPy, and keep in mind that you can use `latex()` command in Sage Math to immediate get the output in a LaTeX friendly print out:

K	J(K;q)
$T(2,3)$	$-q^4+q^3+q$
$T(2,5)$	$-q^7+q^6-q^5+q^4+q^2$
$T(2,7)$	$-q^{10}+q^9-q^8+q^7-q^6+q^5+q^3$
$T(2,9)$	$-q^{13}+q^{12}-q^{11}+q^{10}-q^9+q^8-q^7+q^6+q^4$
$T(2,11)$	$-q^{16}+q^{15}-q^{14}+q^{13}-q^{12}+q^{11}-q^{10}+q^9-q^8+q^7+q^5$
$T(2,13)$	$-q^{19}+q^{18}-q^{17}+q^{16}-q^{15}+q^{14}-q^{13}+q^{12}-q^{11}+q^{10}-q^9+q^8+q^6$
$T(3,4)$	$-q^8+q^5+q^3$
$T(3,5)$	$-q^{10}+q^6+q^4$

Look at `Jonespoly_and_kh_sage.py` for the Python dictionary `TKNOTS14SNAP` for all the Torus knots with less than 14 crossings, and then one can do this to obtain the Jones polynomial:

```
sage: TKNOTS14SNAP[(2,3)].jones_poly()
 q^4 + q^3 + q
```

2. Khovanov Homology implementations: KhoHo and knotkit

2.1. KhoHo. First, install Pari/GP. I recommend using *Homebrew*: `brew install pari`

`KhoHo` was unavailable on the website http://www.geometrie.ch/KhoHo/ and doesn't appear to be on `github`. Dunfield provided `KhoHo-0.9.3.5` by email which I'll try to put on github in the `qsApoly` repository. However, I tried to follow the instructions for KhoHo in the `00README` file, skipping the 'make' command since library files `nicematr.so, print_ranks.so, sparreduce.so` were already there (I tried removing these 3 library files and running make, but 20 errors were generated: see my wordpress blog for the full printout of the errors), but in Pari/GP, by typing `gp` to start up PARI's programmable calculator, and then in `gp`, typing in the command to read KhoHo, this occurred:

```
? read(KH)
  ***   at top level: read(KH)
  ***                 ^
  ***   in function read: read(KhoHo)
  ***                     ^
  ***   in function read: read(KhoHo_basic)
  ***                     ^
  ***   in function read: kill(print_ranks)
  ***                     ^
  *** kill: can't kill that.
  *** Break loop: type 'break' to go back to GP prompt
break> break
```

I also tried `gp > read(KH)` and obtained an error for the break loop.

Please contact me or let people know what's wrong in this case with KhoHo, or how to get it to run.

2.2. knotkit.

2.2.1. *Installation of knotkit.* I suggest doing a `git clone` of `knotkit` from its github repository:

`git clone https://github.com/cseed/knotkit.git`

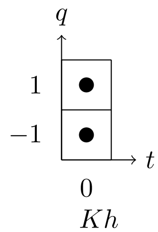
and typing the command 'make' (no need to type './' before, and it should be available to run everywhere on the system) to install and make the executable `kk`.

Use `./kk -h` for the help string.

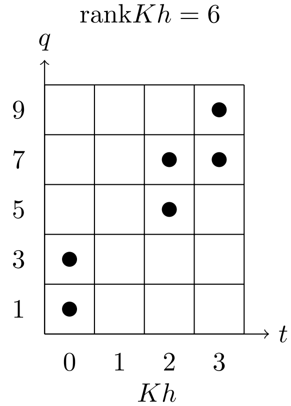
However, if one opens up the code for `kk.cpp` and also run the executable, `kk`, for Khovanov homology (kh) calculations, then the output is in LaTeX; the display is nice, but we want a nice string format so to do manipulations on it. Dunfield provided an alternative version of `kk.cpp` with the option of “`khplainout`” which outputs a string with the categorified Jones polynomial.

2.2.2. *Khovanov Homology computations from knotkit.* Knotkit or `kk`, with the “kh” flag outputs the following, for various torus knots of less than 14 crossings, the graph of the exponents for variables (t, q) as dots:

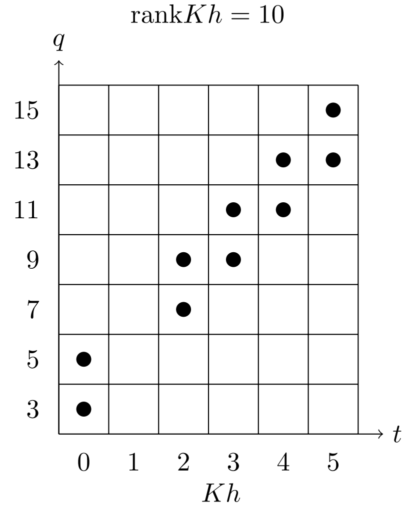
Kh $= Kh(\texttt{T(2,1)};\texttt{Z2})$:
rank$Kh = 2$



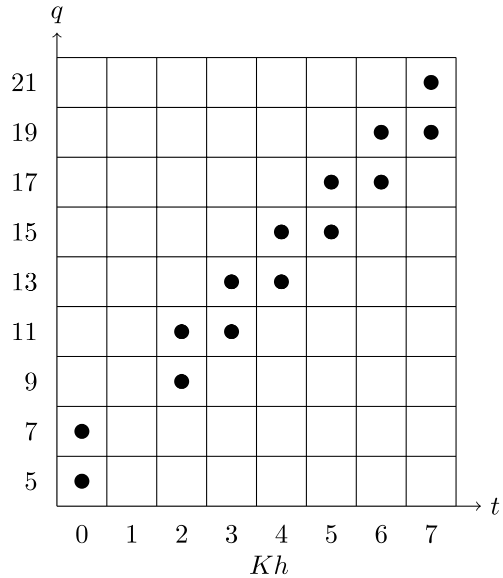
Kh $= Kh(\texttt{T(2,3)};\texttt{Z2})$:
rank$Kh = 6$



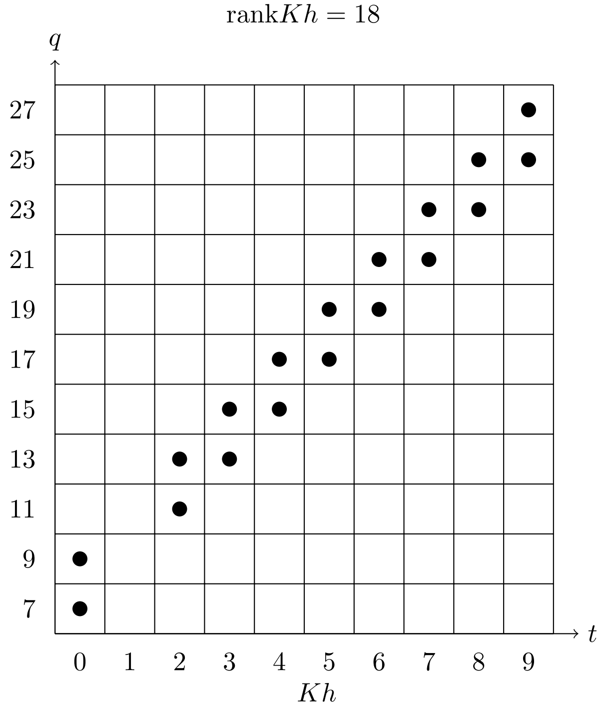
Kh $= Kh(\texttt{T(2,5)};\texttt{Z2})$:
rank$Kh = 10$



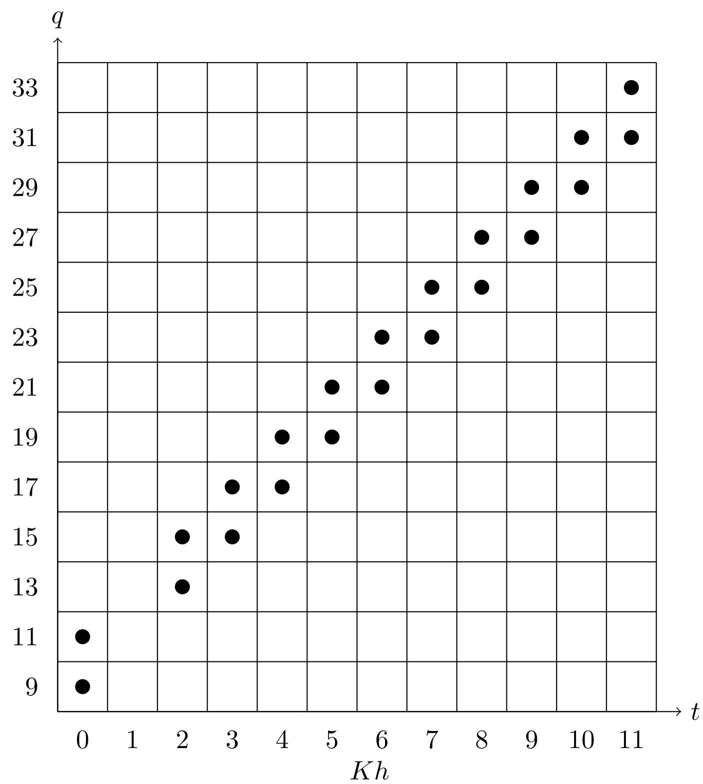
Kh $= Kh(\texttt{T(2,7)};\texttt{Z2})$:
rank$Kh = 14$



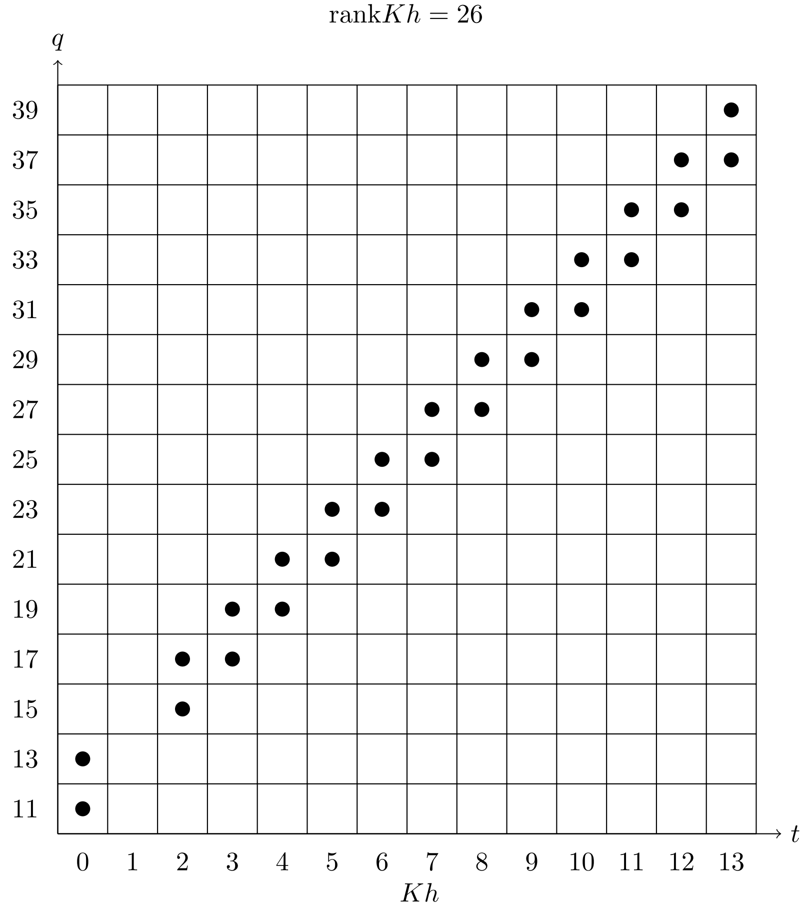
Kh $= Kh($`T(2,9)`$;$ `Z2`$)$:
rank$Kh = 18$



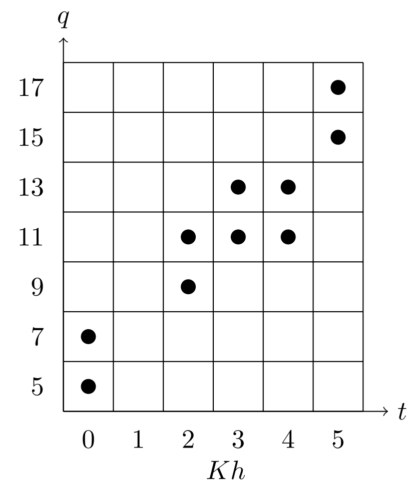
Kh $= Kh($`T(2,11)`$;$ `Z2`$)$:
rank$Kh = 22$



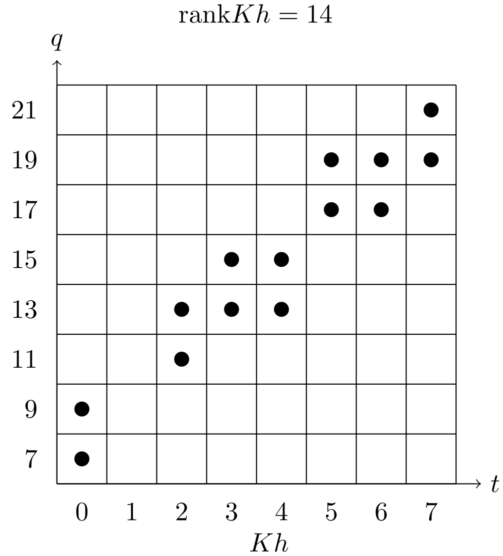
Kh $= Kh($`T(2,13)`$;$`Z2`$)$:
rank$Kh = 26$



Kh $= Kh($`T(3,4)`$;$`Z2`$)$:
rank$Kh = 10$



Kh $= Kh(\texttt{T(3,5)}; \texttt{Z2})$:
rank$Kh = 14$



If you wanted to *save the output* from option flag "khplainout", then one simple and direct manner is to "pipe" the screen output with the symbol ">" at the Command prompt and specify the target file you'd want to save the output. For example, for the Torus knot $T(2,5)$, saving into the file `kh_T0205` in the directory above the current working directory,

```
./kk khplainout -v ‘‘T(2,5)’’ > ../kh_T0205
```

I've done this for all torus knots of less than 14 crossings and saved the output and placed them in the github repository `qsApoly` so others can have access to the Khovanov homology computation outputs readily; they are named files `kh_T0203`, `kh_T0205`, ... in `kk_kh_output`.

If you don't want to do this one-by-one, but a batch of these, one at a time, see the Python script `kk_kh_batch.py`, which you can run directly by typing at the command prompt `python kk_kh_batch.py` or manually, by opening up an interactive shell (i.e. `python -i kk_kh_batch.py`) and then running manually the Python function `direct_output`.

To "clean" or "scrape" the text (strings) of "categorified" Jones polynomial, to get it ready to use for `Sage Math`, I provide the file `kh_scrape.py`. It contains the function `scrape_file` that will take a file containing a *single* categorified Jones polynomial and output a "cleaned up" text (string) for the polynomial, in a Python dictionary, that can be used in Sage Math (just then apply the Sage Math function `sage_eval`). For example:

```
from kh_scrape import scrape_file
Qkh.<t,q> = PolynomialRing(RationalField(),2)
T0203stf = scrape_file('kh_T0203')
T0203poly = sage_eval( T0203stf['poly'], locals={'x1':t,'x2':q} )
T0203poly.substitute(t=1) # for some reason, subs doesn't work in this case
```

Also, from running `kh_scrape.py` and function `scrape_bat` in *Sage Math* (make sure you're in the appropriate working directory)

```
sage: from kh_scrape import scrape_bat
sage: Tknots14 = scrape_bat('khTknotsless14')
sage: Tknots14sage = [ sage_eval(line, locals={'x1':t,'x2':q}) for line in Tknots14 ]
sage: for K in Tknots14sage:
....:         print latex(K)
```

Then you'll have all the torus knots categorified Jones polynomials from Khovanov Homology in a single list "Tknots14sage" (but you'll have to manually keep track of which entry corresponds to which torus knot).

The result of printing with `latex` command in the last command immediately above is this table of "categorified" Jones polynomials from Khovanov Homology (source:`knotkit`)

K	J(K;t,q)
$T(2,1)$	$\frac{q^2+1}{q}$
$T(2,3)$	$t^3q^9 + t^3q^7 + t^2q^7 + t^2q^5 + q^3 + q$
$T(2,5)$	$t^5q^{15} + t^5q^{13} + t^4q^{13} + t^4q^{11} + t^3q^{11} + t^3q^9 + t^2q^9 + t^2q^7 + q^5 + q^3$
$T(2,7)$	$t^7q^{21} + t^7q^{19} + t^6q^{19} + t^6q^{17} + t^5q^{17} + t^5q^{15} + t^4q^{15} + t^4q^{13} + t^3q^{13} + t^3q^{11} + t^2q^{11} + t^2q^9 + q^7 + q^5$

For $T(2,9)$,

$$t^9q^{27} + t^9q^{25} + t^8q^{25} + t^8q^{23} + t^7q^{23} + t^7q^{21} + t^6q^{21} + t^6q^{19} + t^5q^{19} + t^5q^{17} + t^4q^{17} + t^4q^{15} + t^3q^{15} + t^3q^{13} + t^2q^{13} + t^2q^{11} + q^9 + q^7 \tag{1}$$

For $T(2,11)$,

$$\begin{aligned} &t^{11}q^{33} + t^{11}q^{31} + t^{10}q^{31} + t^{10}q^{29} + t^9q^{29} + t^9q^{27} + t^8q^{27} + t^8q^{25} + t^7q^{25} + t^7q^{23} + t^6q^{23} \\ &+ t^6q^{21} + t^5q^{21} + t^5q^{19} + t^4q^{19} + t^4q^{17} + t^3q^{17} + t^3q^{15} + t^2q^{15} + t^2q^{13} + q^{11} + q^9 \end{aligned} \tag{2}$$

For $T(2,13)$,

$$\begin{aligned} &t^{13}q^{39} + t^{13}q^{37} + t^{12}q^{37} + t^{12}q^{35} + t^{11}q^{35} + t^{11}q^{33} + t^{10}q^{33} + t^{10}q^{31} + t^9q^{31} + t^9q^{29} + t^8q^{29} + t^8q^{27} + t^7q^{27} \\ &+ t^7q^{25} + t^6q^{25} + t^6q^{23} + t^5q^{23} + t^5q^{21} + t^4q^{21} + t^4q^{19} + t^3q^{19} + t^3q^{17} + t^2q^{17} + t^2q^{15} + q^{13} + q^{11} \end{aligned} \tag{3}$$

For $T(3,4)$,

$$t^5q^{17} + t^5q^{15} + t^4q^{13} + t^3q^{13} + t^4q^{11} + t^3q^{11} + t^2q^{11} + t^2q^9 + q^7 + q^5 \tag{4}$$

For, $T(3,5)$,

$$t^7q^{21} + t^7q^{19} + t^6q^{19} + t^5q^{19} + t^6q^{17} + t^5q^{17} + t^4q^{15} + t^3q^{15} + t^4q^{13} + t^3q^{13} + t^2q^{13} + t^2q^{11} + q^9 + q^7 \tag{5}$$

The miracle is that if one sets $t = -1$ in the above categorified Jones polynomials, $J(K;t,q)$, then $J(K;t=-1,q) = J(K,q)$, where $J(K,q)$ are the Jones polynomials (modulo conventions), from Table 1!

To show this, keep in mind that for the categorified Jones polynomials, one must *normalize* to the "unknot" which in this case is the $T(2,1)$ torus knot, after setting $t=-1$ (or before? that's my question; please let me know what "dividing" by the unknot means in both cases):

```
sage: normTknots14sage = [K/Tknots14sage[0] for K in Tknots14sage]

# Compare Jones polynomials from SnaPy to Jones polynomials from decategorified Khovanov Homology:
sage: x = var('x')
sage: for i in range(1,len(TKNOTS14SNAP)):
          SnaPyvskh = TKNOTS14SNAP[TKNOTSLIST14[i]].jones_poly().subs(q=x) ==
                              normTknots14sage[i].subs(t= 1).subs(q=sqrt(x))
          print bool( SnaPyvskh )

sage: print "If all True, then J(K;t= 1,q) = J(K;q)!"
```

and indeed it does.

Note that for some reason, the Jones polynomial for $T(2,1)$ of *SnaPy* produces this error:

```
sage: TKNOTS14SNAP[(2,1)].jones_poly()
# stuff
IndexError: list index out of range
```

EY: 20160224: My immediate question is this: what's the convention or normalization that results in SnaPy outputting the Jones polynomial for the torus knots, say $T(2,3)$ trefoil knot, to be

$$-q^4 + q^3 + q$$

vs. the Jones polynomial that results from Khovanov homology, after setting $t = -1$, for the trefoil:

$$-q^8 + q^6 + q^2$$

The latter expression is found on pp. 339 of Bar-Natan's (nicely, pedagogically-friendly) review artcle [2]

3. Exploring the Jones Polynomial and Khovanov Homology: Unanswered Avenues

Armed with our categorified Jones polynomials and Sage Math, there are a number of modules (functions) that can be explored, as seen if one does the `dir()` command on a polynomial (e.g. `dir(Tknots14sage[1])`).

For instance, consider

```
sage: Tknots14sage[1].gradient() # T(2,3)
[3*t^2*q^9 + 3*t^2*q^7 + 2*t*q^7 + 2*t*q^5,
  9*t^3*q^8 + 7*t^3*q^6 + 7*t^2*q^6 + 5*t^2*q^4 + 3*q^2 + 1]
```

which is $\frac{\partial}{\partial t}J(T(2,3);t,q)$ and $\frac{\partial}{\partial q}J(T(2,3);t,q)$. Are there any relationships we can discover between the categorified Jones polynomial and its partial(s) (derivatives)?

One can also use the Sage Math module `newton_polytope` to obtain the Newton Polytope immediately. One does the Newton Polytope tell us about categorified Jones polynomials?

Also, there are many Sage Math modules associated with `newton_polytope` (i.e. do e.g. `dir(Tknots14sage[1].newton_polytope())`), such as `face_lattice`:

```
sage: Tknots14sage[1].newton_polytope().face_lattice().list()
[<>, <0>, <1>, <2>, <3>, <1,2>, <0,1>, <0,3>, <2,3>, <0,1,2,3>]
sage: Tknots14sage[7].newton_polytope().face_lattice().list()
[<>, <0>, <1>, <2>, <3>, <4>, <1,2>, <0,1>, <0,4>, <3,4>, <2,3>, <0,1,2,3,4>]
```

So the face lattice of a $T(2,2j+1)$ torus knot, $j = 1\ldots 6$, is different from the family of torus knots $T(3,4)$ and $T(3,5)$. What does that mean?

This reference page might help with polytopes: A class to keep information about faces of a polyhedron

Something else one could try to do is to repeat Witten's celebrated computation of the Jones polynomial from Chern-Simons theory [3] in *Sage Math*. This would entail an understanding, a grasp, of the Weyl group, in this case, $SU(2)$. I tried looking up topics on the Weyl group and Weyl character in Sage Math (cf. Weyl Group, SL versus GL)

3.1. **Installation of SnapPy.**

3.1.1. *Mac OS X Installation of SnapPy.* First, one should simply download SnapPy.dmg, and then double-click the .dmg file and then drag-and-drog the SnapPy icon into the Applications Folder [1].

However, one would like to take advantage of its integration with `Sage Math` and so here's how to install SnapPy into `Sage`.

(1) Go to or `cd` into the directory where the main program `sage` is in; for example, the directory that `sage`, the *executable file* is in, is

    `/Applications/SageMath-6.10.app/Contents/Resources/sage`

    where `sage` here is a *directory*.
(2) Make sure you have `pip` installed and do this command:

    ```
    ./sage -pip install --no-use-wheel snappy
    ```

    cf. `http://www.math.uic.edu/t3m/SnapPy/installing.html#sage`. It should successfully install.

References

[1] M. Culler, N. M. Dunfield, and J. R. Weeks, SnapPy, a computer program for studying the geometry and topology of 3-manifolds, `http://snappy.computop.org`

[2] Dror Bar-Natan. "On Khovanov's categorification of the Jones polynomial." **Algebraic & Geometric Topology**. Volume 2 (2002) 337–370. arXiv:math/0201043v3 [math.QA]

[3] E. Witten, Commun. Math. Phys. **121**, 351 (1989). doi:10.1007/BF01217730

[1] cf. Installing SnapPy