Python classes, operators, and q-analogues of multiple zeta values

Henrik Bachmann
Nagoya University

Computing Multiple Zeta Seminar, 7th July 2022

Download these slides and code at
`https://zetapedia.miraheze.org/wiki/Computing_Multiple_Zeta_Seminar`.

Overview

Overview of this talk:

- Classes in Sage/Python
- Overloading operators
- Quasi-shuffle products
- Implementation of quasi-shuffle products in Sage
- q-analogues of multiple zeta values + qMZV Calculator (`https://www.henrikbachmann.com/qmzv.html`)

Software I use:

- Until now I used a lot of Pari/GP & Mathematica. Recently, I am interested in Sage.
- For me the most convenient way to use Sage and Pari/GP is by using **JupyterLab** `https://doc.sagemath.org/html/en/reference/spkg/jupyterlab.html`
- Thanks to Hiroses talk we also know that there is a nice way to share Sage programs: **SageMathCell** `https://sagecell.sagemath.org/static/about.html`.

Sage time

We will do an example in Sage now. The code of this example can be found here:
`https://www.henrikbachmann.com/example1.html`

Quasi-shuffle product

- R: commutative ring.
- A: countable set (**alphabet**).
- $\diamond$: commutative and associative product on RA.
- **word**: monic monomial in the non-commutative polynomial ring $R\langle A\rangle$. ($\mathbf{1}$: empty word)

Definition

The **quasi-shuffle product** $*_\diamond$ on $R\langle A\rangle$ is defined as the R-bilinear product satisfying $\mathbf{1} *_\diamond w = w *_\diamond \mathbf{1} = w$ for any word $w \in R\langle A\rangle$ and

$$aw *_\diamond bv = a(w *_\diamond bv) + b(aw *_\diamond v) + (a \diamond b)(w *_\diamond v)$$

for any letters $a, b \in A$ and words $w, v \in R\langle A\rangle$.

The usual examples for multiple zeta values:

- Shuffle product: $R = \mathbb{Q}$, $A = \{x, y\}$, $a \diamond b = 0$.
- Harmonic product: $R = \mathbb{Q}$, $A = \{z_1, z_2, \ldots\}$, $z_{k_1} \diamond z_{k_2} = z_{k_1+k_2}$.

Sage time

We will do an example in Sage now. The code of this example can be found here:
`https://www.henrikbachmann.com/example2.html`

q-analogues of MZV

Define for $k \in \mathbb{Z}$ and $m \geq 1$

$$[m]_q = \frac{1-q^m}{1-q} \underset{q\to 1}{\longrightarrow} m\,, \qquad f_k(m) = \begin{cases} \left(\frac{q^m}{[m]_q}\right)^{-k}, & \text{if } k \leq 0 \\ \frac{q^{(k-1)m}}{[m]_q^k}, & \text{if } k \geq 1 \end{cases} \underset{q\to 1}{\longrightarrow} \frac{1}{m^{|k|}}$$

Definition

For $(k_1,\dots,k_r) \in \mathbb{Z}^r$ with $k_1 \notin \{0,1\}$ we define the ***q*-multiple zeta value**

$$\zeta_q(k_1,\dots,k_r) = \sum_{m_1>\dots>m_r>0} \prod_{j=1}^{r} f_{k_j}(m_j) \quad \underset{q\to 1}{\longrightarrow} \zeta(|k_1|,\dots,|k_r|) \quad (k_1 \neq -1)$$

These can be viewed as element $\mathbb{Q}[[q]]$ or as complex functions in the open unit disc ($|q| < 1$).

- Bradley-Zhao: $k_1,\dots,k_r \in \mathbb{Z}_{\geq 1}$ $\quad (k_1 \neq 1)$
- Takeyama: $k_1,\dots,k_r \in \mathbb{Z}_{\geq 1} \cup \{-1\}$ $\quad (k_1 \neq 1)$
- Schlesinger-Zudilin: $k_1,\dots,k_r \in \mathbb{Z}_{\leq 0}$ $\quad (k_1 \neq 0)$

- Since we have for $k \geq 1$

$$\frac{q^{km}}{[m]_q^k} = (-1)^{k-1}(1-q)^{k-1}\frac{q^m}{[m]_q} + \sum_{j=2}^{k}(-1)^{k-j}(1-q)^{k-j}\frac{q^{(j-1)m}}{[m]_q^j},$$

  we see that the Schlesinger-Zudiling qMZVs span the same space as the Bradley-Zhao-Takeyama qMZVs.

- In the usual setup one considers the spaces

$$\mathfrak{H}^0 = \mathbb{Q} + x\mathfrak{H}y \quad \subset \quad \mathfrak{H}^1 = \mathbb{Q} + \mathfrak{H}y \quad \subset \quad \mathfrak{H} = \mathbb{Q}\langle x, y\rangle\,,$$

  and writes $z_k = x^{k-1}y$, i.e. $\mathfrak{H}^1 = \mathbb{Q}\langle z_1, z_2, \ldots\rangle$.

Instead of working with $\mathbb{Q}$-vector spaces we will work with $\mathcal{C}$-modules, where

$$\mathcal{C} = \mathbb{Q}[\hbar, \hbar^{-1}]\,.$$

Definition

(i) We define $\widehat{\mathfrak{H}} = \mathcal{C}\langle a, b\rangle$ and its two $\mathcal{C}$-submodules

$$\widehat{\mathfrak{H}}^0 = \mathcal{C} + a\widehat{\mathfrak{H}}b \quad \subset \quad \widehat{\mathfrak{H}}^1 = \mathcal{C} + \widehat{\mathfrak{H}}b \quad \subset \quad \widehat{\mathfrak{H}}\,.$$

(ii) For $k \geq 1$ we write

$$e_k = a^{k-1}(a+\hbar)b, \qquad e_{\bar{k}} = a^k b, \qquad e_{\bar{0}} = b = \frac{1}{\hbar}(e_1 - e_{\bar{1}})$$

With this we have

$$\widehat{\mathfrak{H}}^1 = \mathcal{C}\langle e_{\bar{1}}, e_1, e_2, e_3, \dots\rangle = \mathcal{C}\langle e_{\bar{0}}, e_{\bar{1}}, e_{\bar{2}}, \dots\rangle,$$

$$\widehat{\mathfrak{H}}^0 = \mathcal{C} + \langle e_{k_1} \dots e_{k_r} \mid r \geq 1, k_1, \dots, k_r \in \mathbb{Z}_{\geq 1} \cup \{-1\}, k_1 \neq 1\rangle_{\mathcal{C}}\,.$$

Algebraic setup for q-analogues of MZV: Harmonic product

View $\mathbb{Q}[[q]]$ as a $\mathcal{C}$-module by defining the multiplication of $\hbar$ as the multiplication with $(1-q)$. Define the $\mathcal{C}$-linear map

$$\begin{aligned}\zeta_q : \widehat{\mathfrak{H}}^0 &\longrightarrow \mathbb{Q}[[q]] \\ e_{k_1} \dots e_{k_r} &\longmapsto \zeta_q(k_1, \dots, k_r)\,.\end{aligned}$$

We define $*_q = *_\diamond$ to be the quasi-shuffle product on $\widehat{\mathfrak{H}}^1 = \mathcal{C}\langle A \rangle$ defined by the alphabet $A = \{e_{\bar{1}}, e_1, e_2, \dots\}$ and the product $\diamond$ on $\mathcal{C}A$ given by $(k, k_1, k_2 \geq 1)$

$$e_{\bar{1}} \diamond e_{\bar{1}} = e_2 - \hbar e_{\bar{1}}, \qquad e_{\bar{1}} \diamond e_k = e_{k+1}, \qquad e_{k_1} \diamond e_{k_2} = e_{k_1+k_2} + \hbar e_{k_1+k_2-1}\,.$$

Notice that we have $e_{\overline{k_1}} \diamond e_{\overline{k_1}} = e_{\overline{k_1+k_2}}$ for any $k_1, k_2 \geq 0$. This gives a commutative $\mathcal{C}$-algebra $\widehat{\mathfrak{H}}^1_{*_q}$.

Proposition

The map $\zeta_q : \widehat{\mathfrak{H}}^0_{*_q} \to \mathbb{Q}[[q]]$ is an $\mathcal{C}$-algebra homomorphism.

Algebraic setup for q-analogues of MZV: Shuffle product

Definition

We define the q-shuffle product $⧢_q$ on $\widehat{\mathfrak{H}}$ as the $\mathcal{C}$-bilinear product, which satisfies $1 ⧢_q w = w ⧢_q 1 = w$ for any $w \in \widehat{\mathfrak{H}}$ and

$$aw ⧢_q av = a((aw) ⧢_q v + w ⧢_q (av) + \hbar w ⧢_q v)\,,$$
$$(bw) ⧢_q v = w ⧢_q (bw) = b(w ⧢_q v)$$

for words $w, v \in \widehat{\mathfrak{H}}$.

This gives a commutative $\mathcal{C}$-algebra $\widehat{\mathfrak{H}}_{⧢_q}$ with subalgebra $\widehat{\mathfrak{H}}^0_{⧢_q}$.

Proposition

The map $\zeta_q : \widehat{\mathfrak{H}}^0_{⧢_q} \to \mathbb{Q}[[q]]$ is an $\mathcal{C}$-algebra homomorphism.

We obtain double shuffle relations: For all $w, v \in \widehat{\mathfrak{H}}^0$ we have $\zeta_q(w ⧢_q v - w *_q v) = 0$.

Algebraic setup for q-analogues of MZV: Duality

Define the anti-automorphism $\sigma : \widehat{\mathfrak{H}} \to \widehat{\mathfrak{H}}$ by $\sigma(1) = 1$ and

$$\sigma(a) = \hbar b, \qquad \sigma(b) = \hbar^{-1} a\,.$$

Proposition

- *For all $w \in \widehat{\mathfrak{H}}^0$ we have*

$$\zeta_q(\sigma(w)) = \zeta_q(w).$$

- *For all $w, v \in \widehat{\mathfrak{H}}^0$ we have*

$$w \sqcup\!\sqcup_q v = \sigma(\sigma(w) *_q \sigma(v))\,.$$

Conjecture

All algebraic/linear relations among ζ_q are a consequence of the harmonic product $*_q$ and the σ-invariance.

An implementation of the ζ_q can be found here: https://www.henrikbachmann.com/qmzv.html

---

Calculate the first 15 coefficients of $\zeta_q(2,3,4)$:

```
qmzv(2,3,4).q(15)
```

Output:

```
q^10 - 4*q^11 + 9*q^12 - 13*q^13 + 11*q^14 - 2*q^15 + O(q^16)
```

Calculate $\zeta_q(2)\zeta_q(3)$ with the harmonic product $*_q$:

```
qmzv(2) * qmzv(3)
```

Output:

```
ζ_q(2,3)+ζ_q(3,2)+ζ_q(5)+(h)*ζ_q(4)
```

Calculate $\zeta_q(2)\zeta_q(3)$ with the shuffle product $ш_q$:

```
qmzv(2) % qmzv(3)
```

Output:

```
(h)*ζ_q(2,2)+(2*h)*ζ_q(3,-1)+(3)*ζ_q(3,2)+(3)*ζ_q(4,-1)+(2*h)*ζ_q(3,1)+(3)*ζ_q
    (4,1)+(1)*ζ_q(2,3)
```

qMZV Calculator: Examples

Switch between models (BZT or SZ)

```
qmzv(3,2).toSZ()
```

Output:

```
ζ_q(-3,-2)+(h)*ζ_q(-3,-1)+(h)*ζ_q(-2,-2)+(h^2)*ζ_q(-2,-1)
```

```
qmzv(3,-2).toBZT()
```

Output:

```
(-h)*ζ_q(3,-1)+(1)*ζ_q(3,2)
```

Calculating $\zeta_q(\sigma(e_3))$

```
qmzv(3).sigma()
```

Output:

```
(h^2)*ζ_q(-1,0,0)+(h^2)*ζ_q(-1,0)
```

Checking $w \sqcup\!\sqcup_q v = \sigma(\sigma(w) *_q \sigma(v))$ for $w = e_3$,$v = e_2 e_1$

```
(qmzv(3) % qmzv(2,1) - (qmzv(3).sigma() * qmzv(2,1).sigma()).sigma() ).
    toBZT()
```

Output:

```
0
```

qMZV Calculator: qlindep

To find relations among q-analogues one can use the function `qlindep`:

```
rel=qlindep([qmzv(3),qmzv(4),qmzv(2,2),qmzv(3,1),qmzv(2,1,1),qmzv(2,1)])
for r in rel:
    print(r)
```

Output:

```
ζ_q(3)-ζ_q(2,1)
ζ_q(4)-ζ_q(2,1,1)
ζ_q(2,2)+ζ_q(3,1)-ζ_q(2,1,1)
```

qMZV Calculator: MZV

Multiple zeta values are also implemented (Student project by Masataka Satoh). Here are some of the implemented functions:

```
print( mzv(2) * mzv(3) ) # harmonic product
print( mzv(2) % mzv(3) ) # shuffle product
print( mzv(5).toMZV() )  # numerical value
print( mzv(5).dual() ) # dual
print( mzv(5).toHoffmanN()) # writes an element in the Hoffman basis
print( mzv(2,1,1,2,1).simplifyN() ) # writes an element in smallest
    possible depth
print( (mzv(4)-mzv(2,1,1)).simplifyN() )
```

Output:

```
ζ(2,3)+ζ(3,2)+ζ(5)
3*ζ(3,2)+6*ζ(4,1)+ζ(2,3)
1.0369277551433699263313654864570341680570809195019
ζ(2,1,1,1)
4/5*ζ(3,2)+6/5*ζ(2,3)
5*ζ(6,1)+3*ζ(5,2)
0
```