Computing $\zeta(n_1, \ldots, n_r)$ numerically

Explanation of Zagier's approach, and possible extensions

Steven Charlton
Universität Hamburg

12 May 2022

Computing Multiple Zeta Seminar

Outline

1 Introduction and motivation

2 Numerical evaluation – truncated series

3 Numerical Evaluation – asymptotic tail

4 Extensions
- Schur MZV's
- Alternating MZV's
- Multiple t values

References

Source: https://www.newton.ac.uk/seminar/3542/
Section starting at 11m50s until 22m00s

Talk by Zagier during the Grothendieck-Teichmüller groups, deformation and operads programme at the Newton Institute, 27 March 2013

Also: “Standard and less standard asymptotic methods”
Lecture course by Zagier at SISSA/IGAP/SUSTech, Spring 2022
Videos on YouTube
https://youtube.com/playlist?list=PLLq_gUfXAnknpvW3cegUx8ec9KN0mGZA2

Multiple zeta values

Definition (MZV's)

The multiple zeta value $\zeta(s_1, \ldots, s_r)$ is defined by

$$\zeta(s_1, \ldots, s_r) = \sum_{k_1 > k_2 > \ldots > k_r \geq 1} \frac{1}{k_1^{s_1} \cdots k_r^{s_r}}$$

Question

How (and why) to numerically evaluate?

Why?

- Experimentation (to discover conjectures)
- Evidence (to further support conjectures)
- Verification (while developing proofs)
- Hand-on feeling (to understand other theorems)

Examples

Routine `zetamult([s1,...,sr],{t=0})` in gp/pari ($\geq$v2.13.0) to evaluate $\zeta(s_1, \ldots, s_r)$

```
zetamult([3,1])
%1 = 0.27058080842778454787900092413529197569
% / zeta(4)       \\ % = previous result.  Can use zeta(4) or zetamult([4])
%2 = 0.25000000000000000000000000000000000000
bestappr(%)       \\ find rational approximation to decimal
%3 = 1/4
```

So $\zeta(3,1) = \dfrac{1}{4}\zeta(4) = \dfrac{\pi^4}{360} = \dfrac{\pi^4}{3 \cdot 5!}$ (numerically, to many decimal places)

Some experimentation later

```
bestappr(zetamult([3,1,3,1]) / Pi^8)
%4 = 1/1814400           \\ = 5 * 9!
bestappr(zetamult([3,1,3,1,3,1]) / Pi^12)
%5 = 1/43589145600           \\ = 7 * 13!
```

Conjecture (Zagier) + Theorem (Broadhurst)

$$\zeta(\underbrace{\{3,1\}^n}_{3,1,\ldots,3,1 \text{ repeated } n \text{ times}}) = \frac{\pi^{4n}}{(2n+1)\cdot(4n+1)!}$$

Do it yourself?

So why implement it ourselves?

- Someone has to do it first, what if `zetamult` didn't exist?
- Verification of existing routines (limits, issues, edge cases)
  - Even better with alternative method
- Understand the technique in order to generalise it
  - Schur MZV's
  - Alternating MZV's
  - Multiple t values
- Implementations in other software
  - Mathematica(!)

Numerical evaluation

truncated series

Preparation, general

Naïve approach, just sum the series to a 'high-enough' bound

$$\zeta_M(s_1, \ldots, s_r) := \sum_{M \geq k_1 > k_2 > \cdots > k_r \geq 1} \frac{1}{k_1^{s_1} \cdots k_r^{s_r}}$$

Syntax in gp/pari

$$\sum_{i=a}^{b} f(i) \leftrightarrow$$ `sum(i = a, b, f(i))`

$\leftrightarrow$ `ttl = 0; for(i = a, i <= b, i++, ttl += f(i)); ttl`

Syntax in Mathematica

$$\sum_{i=a}^{b} f(i) \leftrightarrow$$ `Sum[f[i], {i, a, b}]`

$\leftrightarrow$ `ttl = 0; For[i = a, i <= b, i++, ttl += f[i]]; ttl`

Interlude, Precision in Mathematica

Every real number carries its own precision:

```
> Precision[1.07289]
MachinePrecision
> Precision[1.0000000000000000000000000000001]
31
```

After arithmetic, Mathematica produces a number with maximum possible valid precision, given the input.

Trick to calculate emulate \p 50 in Mathematica

```
> $MinPrecision = 50;
> 1.0 (* doesn't default to higher precision, use 1.0`1
       (read: 1.0 to precision 1) to make explicit *)
1.
> 1.0`1
1.00000000000000000000000000000000000000000000000000000
> SetPrecision[1.0, 6] (* Gives warning about less than $MinPrecision *)
SetPrecision::precsm: Requested precision 6 is smaller than $MinPrecision.
    Using $MinPrecision instead.
1.00000000000000000000000000000000000000000000000000000
> SetPrecision[1.0, 65]
(* Can always use higher precision *)
1.0000000000000000000000000000000000000000000000000000000000000000000000000
```

Preparation, depth 2 implementation

We have

$$\zeta_M(s_1, s_2) = \sum_{i_1=1}^{M} \left(\sum_{i_2=1}^{i_1-1} \frac{1}{i_1^{s_1} i_2^{s_2}} \right)$$

gp/pari implementation

```
zetaDbl(M, s) = { sum(i1 = 1, M, sum(i2 = 1, i1 - 1,
                         1.0 / (i1^s[1] * i2^s[2]) ))}
```

Mathematica implementation

```
zetaDbl[M_, s_List] := Sum[1.0`1 / (i1^s[[1]] * i2^s[[2]]),
                              {i1, 1, M}, {i2, 1, i1-1}]
```

depth 2 results

	Time	Result	Accuracy (Result $-\zeta(s1,s2)$)
`zetaDbl(10, [2,1])`	<1ms	$0.8303661265\ldots$	$-0.3716907766\ldots$
`zetaDbl(100, [2,1])`	15ms	$1.1405161621\ldots$	$-0.0615407409\ldots$
`zetaDbl(1000, [2,1])`	.35s	$1.1935759233\ldots$	$-0.0084809798\ldots$
`zetaDbl(10000, [2,1])`	35s	$1.2009781989\ldots$	$-0.0010787041\ldots$

`Mathematica` similar (some extra overhead)

Know $\zeta(2,1) = \zeta(3) = 1.2020569031\ldots$, what outcomes do we see.

Analysis

- Accuracy is very poor
- Runtime is very long
  - Algorithm is $O(M^2)$

For $\zeta_M(s_1, s_2, s_3)$ would be $O(M^3)$, *even slower*. Would use three nested loops/sums.

Question

How to even write function for general depth d? *Without* variable number of loops?

Efficiency with recursion

Let's note the following

$$\begin{aligned}\zeta_M(s_1,\dots,s_r) &= \sum_{i_1=1}^{M} \zeta_{i_1-1}(s_2,\dots,s_r)\cdot\frac{1}{i_1^{s_1}}\\ &= \sum_{i_1=1}^{M-1} \zeta_{i_1-1}(s_2,\dots,s_r)\cdot\frac{1}{i_1^{s_1}} + \zeta_{M-1}(s_2,\dots,s_r)\cdot\frac{1}{M^{s_1}}\\ &= \zeta_{M-1}(s_1,\dots,s_r) + \zeta_{M-1}(s_2,\dots,s_r)\frac{1}{M^{s_1}}\end{aligned}$$

So, knowing

$$[\zeta_{M-1}(s_1,\dots,s_r),\zeta_{M-1}(s_2,\dots,s_r),\dots,\zeta_{M-1}(s_r),\overbrace{\zeta_{M-1}(\emptyset)}^{:=1}]$$

calculate

$$[\zeta_{M}(s_1,\dots,s_r),\zeta_{M}(s_2,\dots,s_r),\dots,\zeta_{M}(s_r),\underbrace{\zeta_{M-1}(\emptyset)}_{:=1}]$$

Start with $[0,\dots,0,1] =: [\zeta_0(s_1,\dots,s_r),\dots,\zeta_0(s_r),\zeta_0(\emptyset)]$

Algorithm can be $O(Mr)$ now, much better! (Depending on implementation!)

Algorithm with recursion

Key equations:

$$\zeta_i(s_1, \ldots, s_r) = \zeta_{i-1}(s_1, \ldots, s_r) + \zeta_{i-1}(s_2, \ldots, s_r)\frac{1}{i^{s_1}}$$

$$\zeta_M(\emptyset) = 1\,, \quad \zeta_0(s_1, \ldots, s_r) = \begin{cases} 0 & \text{if } r > 0 \\ 1 & \text{if } r = 0 \end{cases}$$

gp/pari implementation

```
zetaM(M, s) = {
    if(length(s) == 0, return(1.0));
    if(M == 0, return(0));
    zetaM(M-1, s) + zetaM(M-1, s[2..length(s)]) * 1/M^s[1] }
```

Mathematica implementation

```
zetaM[_, {}] := 1.0`1;
zetaM[0, s_List] := 0;
zetaM[M_, s_List] :=  zetaM[M - 1, s]
                           + zetaM[M - 1, s[[2 ;; All]]] / M^s[[1]];
```

Works, but still very slow. Memory and CPU overhead in calling functions (plus finite stack).

Warning, many redundant calculations

$\zeta_M(s_1, s_2)$ needs $\zeta_{M-1}(s_1, s_2)$, so $\zeta_{M-2}(s_2)$. Also needs $\zeta_{M-1}(s_2)$, which needs $\zeta_{M-2}(s_2)$.

Algorithm with recursion as loop

Key equation: $$\zeta_i(s_1,\ldots,s_r) = \zeta_{i-1}(s_1,\ldots,s_r) + \zeta_{i-1}(s_2,\ldots,s_r)\frac{1}{i^{s_1}}$$

Mathematica implementation

```
zetaM[M_, s_List] := Module[{vec, i, r}, (
   vec = Join[ Table[0, Length[s]], {1.0`1}];
   For[i = 1, i <= M, i++,
      For[r = 1, r <= Length[s], r++,
         vec[[r]] = vec[[r]] + vec[[r + 1]]*1/i^s[[r]]
      ]; ];
   vec[[1]] )];
```

gp/pari implementation

```
zetaM(M,s) = {
  vec = concat(vector(length(s)), [1.0]);
    for(i = 1, M,
      for(r = 1, length(s),
         vec[r] = vec[r] + vec[r+1] * 1/i^s[r]
    ));
  vec[1] }
```

Results

	Time	Result	Accuracy (Result $-\zeta(s_1,s_2)$)
`zetaM(10, [2,1])`	<1ms	$0.8303661265\ldots$	$-0.3716907766\ldots$
`zetaM(1000, [2,1])`	15ms	$1.1935759233\ldots$	$-0.0084809798\ldots$
`zetaM(100000, [2,1])`	220ms	$1.2019260023\ldots$	$-0.0001309007\ldots$
`zetaM(10^6, [2,1])`	2s	$1.2020415104\ldots$	$-1.5392718776\ldots\times 10^{-5}$
`zetaM(10^7, [2,1])`	20s	$1.2020551336\ldots$	$-1.7695310456\ldots\times 10^{-6}$
`zetaM(10^5, [3,1,3,1])`	390ms	$0.0052295694\ldots$	$-1.4440791284\ldots\times 10^{-10}$
`zetaM(10^6, [3,1,3,1])`	3.4s	$0.0052295695617\ldots$	$-1.7556092786\ldots\times 10^{-12}$
`zetaM(10^7, [3,1,3,1])`	37s	$0.0052295695635\ldots$	$-2.0671284752\ldots\times 10^{-14}$

Analysis

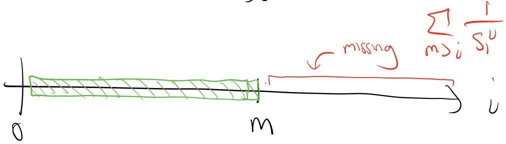
- Accuracy is still poor
- Runtime is better
  - Algorithm is $O(Mr)$

Numerical Evaluation

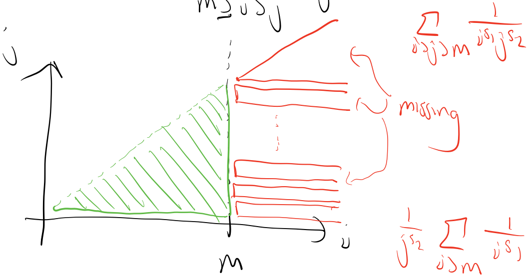
asymptotic tail

What's missing from $\zeta_M(s_1, \dots, s_r)$?

$$\zeta_m(s_1) = \sum_{m \ge i} \frac{1}{s_1^{i}}$$



$$\zeta_m(s_1, s_2) = \sum_{m \ge i > j} \frac{1}{i^{s_1} j^{s_2}}$$



Asymptotic for $\zeta(s_1, \ldots, s_r)$ tail

Goal

Find and incorporate approximation for

$$\zeta_{\gg M}(s_1, \ldots, s_r) \coloneqq \sum_{i_1 > \cdots > i_r > M} \frac{1}{i_1^{s_1} \cdots i_r^{s_r}}$$

Recall

Theorem (Euler-Maclaurin Summation)

$$\sum_{n=a}^{b} f(n) = \int_a^b f(x)\mathrm{d}x + \frac{f(a)+f(b)}{2} + \sum_{k=1}^{N-1} \frac{B_{k+1}}{(k+1)!}(f^{(k)}(b) - f^{(k)}(a))$$
$$- \frac{(-1)^N}{N!} \int_a^b B_N(x - \lfloor x \rfloor) f^{(N)}(x)\mathrm{d}x$$

Can argue/show that $\zeta_{\gg M}(s_1, \ldots, s_r)$ is polynomial (power series) in $1/M$, plus terms which go *rapidly* to 0 as $M \to \infty$. (Details omitted!)

Asymptotic for $\zeta(s_1, \ldots, s_r)$ tail

Depth 1 (c.f. last time, Tasaka's talk)

$$\zeta_{\gg M}(s) = \zeta(s) - \zeta_M(s) = \frac{1}{s-1}\frac{1}{M^{s-1}} - \frac{1}{2M^s} + \sum_{k=1}^{N} \frac{B_{k+1}}{k+1}\binom{s+k-1}{k}\frac{1}{M^{s+k}} \underbrace{- \binom{s+N-1}{N}\int_a^\infty B_N(x - \lfloor x \rfloor) x^{-s-N} \mathrm{d}x}_{\to 0 \text{ rapidly}}$$

Recursively construct higher depth?

Recursion for $\zeta_{\gg M}$

$$\begin{aligned}
\zeta_{\gg M-1}(s_1, \ldots, s_r) &= \sum_{k_1 > \cdots > k_r > M-1} \frac{1}{k_1^{s_1} \cdots k_r^{s_r}} \qquad \text{(so: } k_r > M \text{ or } k_r = M\text{)} \\
&= \sum_{k_1 > \cdots > k_r > M} \frac{1}{k_1^{s_1} \cdots k_r^{s_r}} + \sum_{k_1 > \cdots > k_{r-1} > M} \frac{1}{k_1^{s_1} \cdots k_{r-1}^{s_{r-1}}} \cdot \frac{1}{M^{s_r}} \\
&= \zeta_{\gg M}(s_1, \ldots, s_r) + \frac{1}{M^{s_r}} \zeta_{\gg M}(s_1, \ldots, s_{r-1})
\end{aligned}$$

Goal: asymptotic tail to approximate $\zeta_{\gg M}(s_1, \ldots, s_r)$ then evaluate $\zeta_{\gg 0}(s_1, \ldots, s_r) = \zeta(s_1, \ldots, s_r)$

Recursion for tail

Assume/since $\zeta_{\gg M}(s_1,\ldots,s_r) \sim A_{s_1,\ldots,s_r}(M^{-1})$ for $A_{s_1,\ldots,s_r}(x)$ some power-series. Recursion

$$\zeta_{\gg M-1}(s_1,\ldots,s_r) = \zeta_{\gg M}(s_1,\ldots,s_r) + \frac{1}{M^{s_r}}\zeta_{\gg M}(s_1,\ldots,s_{r-1})$$

implies

$$A_{s_1,\ldots,s_r}\left(\frac{x}{1-x}\right) = A_{s_1,\ldots,s_r}(x) + x^{s_r}A_{s_1,\ldots,s_{r-1}}(x)$$

If we know $A_{s_1,\ldots,s_{r-1}}$ up to order x^N, this sets up a system of equations for $A_{s_1,\ldots,s_r}(x)$. This gives $A_{s_1,\ldots,s_r}(x)$ up to order x^N. Since $\zeta_{\gg M}(\emptyset) = 1$, $A_\emptyset(x) \equiv 1$ starts the recursion.

Write $A(x) = \sum_{i=0}^\infty a_i x^i$, $B(x) = \sum_{i=0}^\infty b_i x^i$, then $A(\frac{x}{1-x}) - A(x) = x^\alpha B(x)$ gives

$$\sum_{k=0}^\infty \sum_{j=1}^\infty a_k \binom{k+j-1}{j} x^{k+j} = \sum_{k=\alpha} b_{k-\alpha}x^k \implies \sum_{k=0}^{i-1} a_k \binom{i-1}{i-k} = b_{i-\alpha}$$

Hence

$$a_i = \frac{1}{i}\left\{b_{i+1-\alpha} - \sum_{k=0}^{i-1} a_k \binom{i}{i+1-k}\right\}$$

Algorithm for asymptotic series

With given s $= (s_1, \ldots, s_r)$ and trorder $\in \mathbb{Z}_{>0}$ the goal for asymptotic series order. Variable asym[i,d] corresponds to x^{i-1} in $A_{s_1,\ldots,s_{d-1}}$

Mathematica implementation

```
asym = Table[0, {Length[s] + 1}, {trorder + 1}];
asym[[1, 1]] = 1;
For[d = 1, d <= Length[s], d++,
  For[i = 1, i <= trorder, i++,
    asym[[d+1, i+1]] = 1/i (
    If[i+1-s[[d]] + 1 < 1, 0, asym[[d-1 + 1, i+1-s[[d]] + 1]]]
    - Sum[asym[[d+1, k+1]] Binomial[i, i+1-k], {k, 0, i-1}] )
]];
```

gp/pari implementation

```
asym = matrix(length(s) + 1, trorder+1);
asym[1,1] = 1;
for(d=1, length(s),
  for(i=1, trorder,
    asym[d+1, i+1] = 1/i * (
    if(i+1-s[d] + 1 < 1, 0, asym[d-1 + 1, i+1-s[d] + 1])
    - sum(k=0, i-1, asym[d+1, k+1] * binomial(i, i+1-k)) )
  ))
```

Evaluation of $\zeta(s_1,\ldots,s_r)$

Assemble ingredients

- $\zeta_{\gg M}(s_1,\ldots,s_r) \approx A_{s_1,\ldots,s_r}(M^{-1})$ (with A truncated to `trorder`, full series diverges).
- For given M and `trorder`, initialise $\zeta_{\gg M}(s_1,\ldots,s_r)$ with $A_{s_1,\ldots,s_r}(M^{-1})$.
- Recursion (as loop) via

$$\zeta_{\gg M-1}(s_1,\ldots,s_r) = \zeta_{\gg M}(s_1,\ldots,s_r) + \frac{1}{M^{s_r}}\zeta_{\gg M}(s_1,\ldots,s_{r-1})$$

- Obtain $\zeta_{\gg 0}(s_1,\ldots,s_r) = \zeta(s_1,\ldots,s_r)$

Implementation in Mathematica

```
zetaM[M_, trorder_, s_List] := Module[{asym, i, d, vec}, (
  asym = Table[0, {Length[s] + 1}, {trorder + 1}];
  asym[[1, 1]] = 1.0`1;

  (* compute asymptotic series *)
  For[d = 1, d <= Length[s], d++,
    For[i = 1, i <= trorder, i++,
      asym[[d + 1, i + 1]] =
        1/i (If[i + 1   < s[[d]], 0, asym[[ d, i - s[[d]] + 2]]]
         - Sum[asym[[d + 1, k + 1]] Binomial[i, i+1 - k], {k, 0, i-1}])
  ]];

  (* initialise recursion *)
  vec = Sum[asym[[All, i + 1]] /M^(i), {i, 0, trorder}];

  (* explicitly sum start of truncated series *)
  For[i = M, i >= 1, i--,
    For[r = Length[s] + 1, r >= 2, r--,
      vec[[r]] = vec[[r]] + vec[[r - 1]]*1/i^s[[r - 1]];
  ]; ];
  Return[vec];
)];
```

Implementation in gp/pari

```
zetaM(M, trorder, s) = {
  asym = matrix(length(s) + 1, trorder+1);
  asym[1,1] = 1.0;

  \\ compute asymptotic series
  for(d=1, length(s),
    for(i=1, trorder,
      asym[d+1, i+1] =
      1/i * (if(i + 1 - s[d] + 1 < 1, 0, asym[d-1 + 1, i+1 - s[d] + 1])
      - sum(k=0, i-1, asym[d+1, k+1] * binomial(i, i+1-k)) )
  ));

  \\ initialise recursion
  vec = sum(i=0, trorder, asym[,i+1]/M^i);

  \\ explicitly sum start of truncated series
  for(i=0, M-1,
    for(r = 0, length(s)-1,
      vec[length(s)+1 - r] = vec[length(s)+1 - r] +
        vec[length(s)+1 - (r+1)] * 1/(M-i)^s[length(s)+1 - (r+1)];
  ));
  return(vec);
  };
```

Results

Optimisations

Can further optimize

- Computing M^i as $M^{i-1} \times M$
- Reusing memory for `asymp`
  - Only need to store 1 expansion at a time?
- Can compute multiple (related) MZV's simultaneously

At precision 500

	Time	Accuracy
`zetaM(100, 50, [7,2,3,4,4,8])`	22ms	$-5.5531085623\ldots \times 10^{-81}$
`zetaM(1000, 50, [7,2,3,4,4,8])`	28ms	$-5.3142751863\ldots \times 10^{-132}$
`zetaM(10^4, 50, [7,2,3,4,4,8])`	170ms	$-5.2834398270\ldots \times 10^{-234}$
`zetaM(1000, 200, [7,2,3,4,4,8])`	240ms	$-3.0318425240\ldots \times 10^{-391}$
`zetaM(10^4, 200, [7,2,3,4,4,8])`	380ms	$-3.5484391885\ldots \times 10^{-512}$

Extensions

Schur MZV's

Definition of Schur-like/Graph MZV's

Definition (By example)

$$\zeta\left(\begin{array}{|c|c|c|}\hline a & b & c \\ \hline \multicolumn{1}{c|}{} & d & e \\ \cline{2-3} \multicolumn{2}{c|}{} & f \\ \cline{3-3}\end{array}\right) = \sum_{\substack{m_a \,\leq\, m_b \,\leq\, m_c \\ \wedge \quad\;\; \wedge \\ m_c \,\leq\, m_d \\ \wedge \\ m_f}} \frac{1}{m_a^a\; m_b^b\; m_c^c\; m_d^d\; m_e^e\; m_f^f}\,,$$

More generally

Definition (Graph zeta function)

Let G be an oriented graph (without closed loops), with edges $e(\bullet,\bullet)$ labelled by $\leq$ and $<$, and vertices by $v_i \in \mathbb{Z}_{>0}$. Define

$$\zeta_{\gg M}(G) = \sum_{\substack{M<m_i<\infty \\ m_i<m_j \iff e(v_i,v_j)=`<' \\ m_i<m_j \iff e(v_i,v_j)=`\leq'}} \frac{1}{\prod_i m_i^{v_i}}$$

Recursion for $\zeta_{\gg M}(G)$

Claim: recursion for $\zeta_{\gg M}(G)$ via 'simpler' graphs $\{g_1, \ldots, g_k\}$.

Idea

Search for 'sources': vertices v_i with no incoming edges. Then $v_i > N-1$ implies $v_i = N$ or $v_i > N$, which flows through the graph.

$$\zeta_{\gg M-1}\left(\begin{array}{ccc} a & b & c \\ & d & e \\ & & f \end{array}\right) = \zeta_{\gg M}\left(\begin{array}{ccc} a & b & c \\ & d & e \\ & & f \end{array}\right) + \frac{1}{M^a}\zeta_{\gg M}\left(\begin{array}{cc} b & c \\ d & e \\ & f \end{array}\right) + \frac{1}{M^{a+b}}\zeta_{\gg M}\left(\begin{array}{cc} & c \\ d & e \\ & f \end{array}\right)$$

$$+\frac{1}{M^{a+b+c}}\zeta_{\gg M}\left(\begin{array}{cc} d & e \\ & f \end{array}\right)$$

Example

For $\zeta_{\gg M}\left(\begin{array}{ccc} a & b & c \\ & d & e \\ & & f \end{array}\right)$, we obtain the following: $\zeta_{\gg M-1}(g_i) = \sum_j \frac{1}{M^{m_{ij}}} \zeta_{\gg M}(g_j)$ where

$$\mathcal{G} := \{g_i\} = \left\{ \begin{array}{ccc} a & b & c \\ & d & e \\ & & f \end{array}, \begin{array}{cc} d & e \\ & f \end{array}, \begin{array}{cc} & c \\ d & e \\ & f \end{array}, \begin{array}{cc} b & c \\ d & e \\ & f \end{array}, \begin{array}{c} f \end{array}, \begin{array}{c} e \\ f \end{array}, \begin{array}{c} c \\ e \\ f \end{array}, \emptyset \right\}$$

$$m_{ij} = \begin{pmatrix} 0 & a+b+c & a+b & a & - & - & - & - \\ - & 0 & - & - & d+e & d & - & - \\ - & - & 0 & - & - & c+d & d & - \\ - & b+c & c & 0 & - & - & - & - \\ - & - & - & - & 0 & - & - & f \\ - & - & - & - & e & 0 & - & - \\ - & - & - & - & - & c & 0 & - \\ - & - & - & - & - & - & - & 0 \end{pmatrix}$$

Treat "–" as ∞, so $\frac{1}{M^{m_{ij}}} = 0$.

Upshot

Recursion for start of series, and to construct asymptotic expansion. Numerical evaluation as before.

Extensions

Alternating MZV's

Definition of alternating MZV's

Definition (Alternating MZV)

Let $\varepsilon_1, \ldots, \varepsilon_r \in \{\pm 1\}$. Define

$$\zeta_{\gg M}\begin{pmatrix} s_1, & \ldots, & s_r \\ \varepsilon_1, & \ldots, & \varepsilon_r \end{pmatrix} = \sum_{n_1 > \cdots > n_r > M} \frac{\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}}{n_1^{s_1} \cdots n_r^{s_r}},$$

and put $\zeta\begin{pmatrix} s_1, & \ldots, & s_r \\ \varepsilon_1, & \ldots, & \varepsilon_r \end{pmatrix} = \zeta_{\gg 0}\begin{pmatrix} s_1, & \ldots, & s_r \\ \varepsilon_1, & \ldots, & \varepsilon_r \end{pmatrix}$.

Recurrence relations

$$\zeta_{\gg 2M-2}\begin{pmatrix} s_1, \ldots, s_r \\ \varepsilon_1, \ldots, \varepsilon_r \end{pmatrix} = \zeta_{\gg 2M-1}\begin{pmatrix} s_1, \ldots, s_r \\ \varepsilon_1, \ldots, \varepsilon_r \end{pmatrix} + \zeta_{\gg 2M-1}\begin{pmatrix} s_1, \ldots, s_{r-1} \\ \varepsilon_1, \ldots, \varepsilon_{r-1} \end{pmatrix} \cdot \frac{\varepsilon_r^{2M-1}}{(2M-1)^{s_r}}$$

$$\zeta_{\gg 2M-1}\begin{pmatrix} s_1, \ldots, s_r \\ \varepsilon_1, \ldots, \varepsilon_r \end{pmatrix} = \zeta_{\gg 2M}\begin{pmatrix} s_1, \ldots, s_r \\ \varepsilon_1, \ldots, \varepsilon_r \end{pmatrix} + \zeta_{\gg 2M}\begin{pmatrix} s_1, \ldots, s_{r-1} \\ \varepsilon_1, \ldots, \varepsilon_{r-1} \end{pmatrix} \cdot \frac{\varepsilon_r^{2M}}{(2M)^{s_r}}$$

Identities for asymptotic series.

Assuming

$$A_r(M^{-1}) \sim \zeta_{\gg 2M}\begin{pmatrix} s_1, \ldots, s_r \\ \varepsilon_1, \ldots, \varepsilon_r \end{pmatrix} \text{ and } B_r(M^{-1}) \sim \zeta_{\gg 2M+1}\begin{pmatrix} s_1, \ldots, s_r \\ \varepsilon_1, \ldots, \varepsilon_r \end{pmatrix}$$

find

$$A_r\left(\frac{x}{1-x}\right) = B_r\left(\frac{x}{1-x}\right) + B_{r-1}\left(\frac{x}{1-x}\right) \cdot \frac{\varepsilon_r x^{s_r}}{(2-x)^{s_2}}$$
$$B_r\left(\frac{x}{1-x}\right) = A_r(x) + A_{r-1}(x) \cdot \frac{x^{s_r}}{2^{s_r}}$$

Upshot

Can find recursion $A_r\left(\frac{x}{1-x}\right) - A_r(x) =$ lower depth , and solve for coefficients term by term. Numerical evaluation as before.

Analysis

Computing the asymptotic series is slower now; my implementation involved double sum of truncation order. Could replace with certain hypergeometric function in `Mathematica`

Extensions

Multiple t values

Definitions of multiple t values

Definition (MtV)

Define

$$t_{\gg M}(s_1, \ldots, s_r) = \sum_{n_1 > \cdots > n_r > M} \frac{1}{(2n_1 - 1)^{s_1} \cdots (2n_r - 1)^{s_r}},$$

and put $t(s_1, \ldots, s_r) = t_{\gg 0}(s_1, \ldots, s_r)$.

Recurrence relation

$$t_{\gg M-1}(s_1, \ldots, s_r) = t_{\gg M}(s_1, \ldots, s_r) + t_{\gg M}(s_1, \ldots, s_{r-1}) \cdot \frac{1}{(2M-1)^{s_r}}$$

Asymptotic series identity

$$A_r\left(\frac{x}{1-x}\right) = A_r(x) + A_{r-1}(x) \cdot \frac{x^{s_r}}{(2-x)^{s_r}}$$

Recurrence for asymptotic series

Find

$$A\left(\frac{x}{1-x}\right) = A(x) + B(x) \cdot \frac{x^{s_r}}{(2-x)^{s_r}}$$

leads to

$$a_i = \frac{1}{i}\left\{ \sum_{k=0}^{i+1-s_r} b_{i+1-s_r-k} \binom{s_r+k-1}{k} 2^{-k-s_r} - \sum_{k=0}^{i-1} a_k \binom{i}{i+1-k} \right\}$$

Upshot

Algorithm for coefficients of asymptotic series $A_r(x)$. Numerical evaluation as before.

Implementation in Mathematica

```
tM[M_, trorder_, s_List] := Module[{}, (
  asym = Table[0, {Length[s] + 1}, {trorder + 1}];
  asym[[1, 1]] = 1.0`1;

  (* compute asymptotic series *)
  For[d = 1, d <= Length[s], d++,
    For[i = 1, i <= trorder, i++,
      asym[[d + 1, i + 1]] =
      1/i (Sum[asym[[d-1 + 1, i+1-s[[d]]-k + 1]] *
          Binomial[s[[d]] + k - 1, k]/2^(k + s[[d]]), {k, 0, i+1-s[[d]]}]
          - Sum[asym[[d + 1, k + 1]] Binomial[i, i + 1 - k], {k, 0, i - 1}])
  ]];


  (* initialise recursion / estimate tail of series *)
  vec = Sum[asym[[All, i + 1]] /(M)^(i), {i, 0, trorder}];

  (*  explicitly sum head of series *)
  For[i = M, i >= 1, i--,
    For[r = Length[s] + 1, r >= 2, r--,
      vec[[r]] = vec[[r]] + vec[[r - 1]]*(1)/(2 i - 1)^s[[r - 1]];
  ]];
  vec
)];
```

Implementation in gp/pari

```
tM(M, trorder, s) = {
  asym = matrix(length(s) + 1, trorder+1);
  asym[1,1] = 1.0;

  \\ compute asymptotic series
  for(d=1, length(s),
    for(i=1, trorder,
      asym[d+1, i+1] = 1/i * (sum(k=0, i+1-s[d], asym[d-1+1,i+1-s[d]-k+1]
        * binomial(s[d]+k-1,k) / 2^(k+s[d]))
        - sum(k=0, i-1, asym[d+1, k+1] * binomial(i, i+1-k)) )
  ));

  \\ initialise recursion / estimate tail of series
  vec = sum(i=0, trorder, asym[,i+1]/M^i);

  \\ explicitly sum head of series
  for(i=0, M-1,
    for(r = 0, length(s)-1,
    vec[length(s)+1 - r] = vec[length(s)+1 - r] + vec[length(s)+1 - (r+1)]
      * 1/(2*(M-i)-1)^s[length(s)+1 - (r+1)];
  ));
  return(vec);
}
```

Summary

- Efficient methods to evaluate $\zeta_M(s_1, \ldots, s_r)$
  - Recursion instead of r loops
  - Unroll recursion as a single loop
- Construction of asymptotic series for $\zeta_{\gg M}(s_1, \ldots, s_r)$
  - Recurrence relation for coefficients
  - Loop to compute this series
- Combine to numerically evaluate $\zeta(s_1, \ldots, s_r)$

- Extensions
  - Schur/graph MZV's
  - Alternating MZV's
  - Multiple t values