Computing hypergeometric L-functions in average polynomial time

Kiran S. Kedlaya
(with Edgar Costa and David Roe)

Department of Mathematics, University of California San Diego*
kedlaya@ucsd.edu
http://kskedlaya.org/slides/

Number Theory and Physics
International Centre for Theoretical Physics, Trieste, Italy
June 25, 2024

Supported by NSF (grants DMS-1802161, DMS-2053473) and UC San Diego (Warschawski Professorship), and during the 2023–2024 academic year by IAS and the Simons Foundation.

* The UC San Diego campus occupies unceded ancestral homelands of the Kumeyaay Nation.

Contents

Hypergeometric data

For $\alpha, \beta \in (\mathbb{Q} \cap [0,1))^n$ with $\alpha_i - \beta_j \notin \mathbb{Z}$ for all i, j, there is an irreducible variation of Hodge structures of rank n on $\mathbf{P}^1 \setminus \{0, 1, \infty\}$ for one of whose periods the Picard–Fuchs equation is the hypergeometric diffeq

$$P(\alpha;\beta)(z\frac{d}{dz})(y) = 0, \quad P(\alpha;\beta)(D) := z\prod_{i=1}^{n}(D+\alpha_i) - \prod_{j=1}^{n}(D+\beta_j - 1).$$

The Hodge vector/motivic weight can be read from the **zigzag function**

$$Z_{\alpha,\beta}(x) := \#\{j : \alpha_j \leq x\} - \#\{j : \beta_j \leq x\}.$$

See for instance this example in LMFDB.

Hereafter we assume that α, β are **balanced**,† meaning that the multiplicity of any $\frac{r}{s} \in \mathbb{Q}$ (in lowest terms) depends only on s. LMFDB includes all balanced HG data with $n \leq 10$.

†Otherwise we get motives defined only over some abelian extension of $\mathbb{Q}$.

Hypergeometric data

For $\alpha, \beta \in (\mathbb{Q} \cap [0,1))^n$ with $\alpha_i - \beta_j \notin \mathbb{Z}$ for all i, j, there is an irreducible variation of Hodge structures of rank n on $\mathbf{P}^1 \setminus \{0, 1, \infty\}$ for one of whose periods the Picard–Fuchs equation is the hypergeometric diffeq

$$P(\alpha;\beta)(z\frac{d}{dz})(y) = 0, \quad P(\alpha;\beta)(D) := z\prod_{i=1}^{n}(D+\alpha_i) - \prod_{j=1}^{n}(D+\beta_j - 1).$$

The Hodge vector/motivic weight can be read from the **zigzag function**

$$Z_{\alpha,\beta}(x) := \#\{j : \alpha_j \leq x\} - \#\{j : \beta_j \leq x\}.$$

See for instance this example in LMFDB.

Hereafter we assume that α, β are **balanced**,[†] meaning that the multiplicity of any $\frac{r}{s} \in \mathbb{Q}$ (in lowest terms) depends only on s. LMFDB includes all balanced HG data with $n \leq 10$.

[†]Otherwise we get motives defined only over some abelian extension of $\mathbb{Q}$.

Hypergeometric data

For $\alpha, \beta \in (\mathbb{Q} \cap [0,1))^n$ with $\alpha_i - \beta_j \notin \mathbb{Z}$ for all i, j, there is an irreducible variation of Hodge structures of rank n on $\mathbf{P}^1 \setminus \{0, 1, \infty\}$ for one of whose periods the Picard–Fuchs equation is the hypergeometric diffeq

$$P(\alpha;\beta)(z\frac{d}{dz})(y) = 0, \quad P(\alpha;\beta)(D) := z\prod_{i=1}^{n}(D+\alpha_i) - \prod_{j=1}^{n}(D+\beta_j-1).$$

The Hodge vector/motivic weight can be read from the **zigzag function**

$$Z_{\alpha,\beta}(x) := \#\{j : \alpha_j \le x\} - \#\{j : \beta_j \le x\}.$$

See for instance this example in LMFDB.

Hereafter we assume that α, β are **balanced**,[†] meaning that the multiplicity of any $\frac{r}{s} \in \mathbb{Q}$ (in lowest terms) depends only on s. LMFDB includes all balanced HG data with $n \le 10$.

[†]Otherwise we get motives defined only over some abelian extension of $\mathbb{Q}$.

L-functions

For α, β balanced, this variation of Hodge structures arises from a family of Chow motives $M^{\alpha,\beta}$ over $\mathbb{Q}$.

For any given $z \in \mathbf{P}^1 \setminus \{0, 1, \infty\}$, the motive $M_z^{\alpha,\beta}$ has bad reduction[‡] at these primes:

- **wild** primes p, at which α or β is not in $\mathbb{Z}_{(p)}^n$;
- **tame** primes p, which are not wild but either z or $z - 1$ is not a p-adic unit.

For such z, we obtain an associated L-function; the goal of this talk is to explain some methods for computing these L-functions **at scale**. These will eventually be deployed in LMFDB.

[‡]This is only an upper bound; there can be a "wild" or "tame" prime at which the reduction is actually good.

L-functions

For α, β balanced, this variation of Hodge structures arises from a family of Chow motives $M^{\alpha,\beta}$ over $\mathbb{Q}$.

For any given $z \in \mathbf{P}^1 \setminus \{0, 1, \infty\}$, the motive $M_z^{\alpha,\beta}$ has bad reduction‡ at these primes:

- **wild** primes p, at which α or β is not in $\mathbb{Z}_{(p)}^n$;
- **tame** primes p, which are not wild but either z or $z - 1$ is not a p-adic unit.

For such z, we obtain an associated L-function; the goal of this talk is to explain some methods for computing these L-functions **at scale**. These will eventually be deployed in LMFDB.

‡This is only an upper bound; there can be a "wild" or "tame" prime at which the reduction is actually good.

L-functions

For α, β balanced, this variation of Hodge structures arises from a family of Chow motives $M^{\alpha,\beta}$ over $\mathbb{Q}$.

For any given $z \in \mathbf{P}^1 \setminus \{0, 1, \infty\}$, the motive $M_z^{\alpha,\beta}$ has bad reduction‡ at these primes:

- **wild** primes p, at which α or β is not in $\mathbb{Z}_{(p)}^n$;
- **tame** primes p, which are not wild but either z or $z - 1$ is not a p-adic unit.

For such z, we obtain an associated L-function; the goal of this talk is to explain some methods for computing these L-functions **at scale**. These will eventually be deployed in LMFDB.

‡This is only an upper bound; there can be a "wild" or "tame" prime at which the reduction is actually good.

L-functions

For α, β balanced, this variation of Hodge structures arises from a family of Chow motives $M^{\alpha,\beta}$ over $\mathbb{Q}$.

For any given $z \in \mathbf{P}^1 \setminus \{0, 1, \infty\}$, the motive $M_z^{\alpha,\beta}$ has bad reduction‡ at these primes:

- **wild** primes p, at which α or β is not in $\mathbb{Z}_{(p)}^n$;
- **tame** primes p, which are not wild but either z or $z-1$ is not a p-adic unit.

For such z, we obtain an associated L-function; the goal of this talk is to explain some methods for computing these L-functions **at scale**. These will eventually be deployed in LMFDB.

‡This is only an upper bound; there can be a "wild" or "tame" prime at which the reduction is actually good.

L-functions

For α, β balanced, this variation of Hodge structures arises from a family of Chow motives $M^{\alpha,\beta}$ over $\mathbb{Q}$.

For any given $z \in \mathbf{P}^1 \setminus \{0, 1, \infty\}$, the motive $M_z^{\alpha,\beta}$ has bad reduction[‡] at these primes:

- **wild** primes p, at which α or β is not in $\mathbb{Z}_{(p)}^n$;
- **tame** primes p, which are not wild but either z or $z - 1$ is not a p-adic unit.

For such z, we obtain an associated L-function; the goal of this talk is to explain some methods for computing these L-functions **at scale**. These will eventually be deployed in LMFDB.

[‡] This is only an upper bound; there can be a "wild" or "tame" prime at which the reduction is actually good.

Some motivation for the project

- Refining the (conjectural) formulas for conductor exponents and Euler factors at wild primes (see below).
- Tabulating L-functions of other objects (e.g., some K3 surfaces, some Calabi–Yau threefolds), which in turn has other applications.
- Finding exotic specializations (e.g., where the motive decomposes, or more generally the Mumford–Tate group shrinks).
- Investigating variation across primes in a single L-function (e.g., Newton polygons).
- Providing "big data" to investigate using ML/AI, as in the discovery of **murmurations**.

Some motivation for the project

- Refining the (conjectural) formulas for conductor exponents and Euler factors at wild primes (see below).
- Tabulating L-functions of other objects (e.g., some K3 surfaces, some Calabi–Yau threefolds), which in turn has other applications.
- Finding exotic specializations (e.g., where the motive decomposes, or more generally the Mumford–Tate group shrinks).
- Investigating variation across primes in a single L-function (e.g., Newton polygons).
- Providing "big data" to investigate using ML/AI, as in the discovery of **murmurations**.

Some motivation for the project

- Refining the (conjectural) formulas for conductor exponents and Euler factors at wild primes (see below).
- Tabulating L-functions of other objects (e.g., some K3 surfaces, some Calabi–Yau threefolds), which in turn has other applications.
- Finding exotic specializations (e.g., where the motive decomposes, or more generally the Mumford–Tate group shrinks).
- Investigating variation across primes in a single L-function (e.g., Newton polygons).
- Providing "big data" to investigate using ML/AI, as in the discovery of **murmurations**.

Some motivation for the project

- Refining the (conjectural) formulas for conductor exponents and Euler factors at wild primes (see below).
- Tabulating L-functions of other objects (e.g., some K3 surfaces, some Calabi–Yau threefolds), which in turn has other applications.
- Finding exotic specializations (e.g., where the motive decomposes, or more generally the Mumford–Tate group shrinks).
- Investigating variation across primes in a single L-function (e.g., Newton polygons).
- Providing "big data" to investigate using ML/AI, as in the discovery of **murmurations**.

Some motivation for the project

- Refining the (conjectural) formulas for conductor exponents and Euler factors at wild primes (see below).
- Tabulating L-functions of other objects (e.g., some K3 surfaces, some Calabi–Yau threefolds), which in turn has other applications.
- Finding exotic specializations (e.g., where the motive decomposes, or more generally the Mumford–Tate group shrinks).
- Investigating variation across primes in a single L-function (e.g., Newton polygons).
- Providing "big data" to investigate using ML/AI, as in the discovery of **murmurations**.

Some motivation for the project

- Refining the (conjectural) formulas for conductor exponents and Euler factors at wild primes (see below).
- Tabulating L-functions of other objects (e.g., some K3 surfaces, some Calabi–Yau threefolds), which in turn has other applications.
- Finding exotic specializations (e.g., where the motive decomposes, or more generally the Mumford–Tate group shrinks).
- Investigating variation across primes in a single L-function (e.g., Newton polygons).
- Providing "big data" to investigate using ML/AI, as in the discovery of **murmurations**.

Conductors and bad Euler factors

To "compute" a single hypergeometric L-function, we want the following. (All "recipes" are available in Magma, Sage, and possibly GP/PARI.)

- Gamma factors (i.e., Euler factors at the archimedean place). There is a simple recipe.
- Euler factors and conductor exponents for each **tame** p. There is a simple recipe.
- Euler factors and conductor exponents for each **wild** p. There is a short list of candidates (getting shorter over time...).
- Euler factors for **good** p. We will truncate the Dirichlet series at X^{-s} for some X, which means we need p^a-Frobenius traces for $p^a \leq X$. There is a simple recipe, but efficiency matters!

Some of these are conjectural; but given a **complete** guess for suitably large X, one can numerically check the functional equation.

Conductors and bad Euler factors

To “compute” a single hypergeometric L-function, we want the following. (All “recipes” are available in Magma, Sage, and possibly GP/PARI.)

- Gamma factors (i.e., Euler factors at the archimedean place). There is a simple recipe.
- Euler factors and conductor exponents for each **tame** p. There is a simple recipe.
- Euler factors and conductor exponents for each **wild** p. There is a short list of candidates (getting shorter over time...).
- Euler factors for **good** p. We will truncate the Dirichlet series at X^{-s} for some X, which means we need p^a-Frobenius traces for $p^a \leq X$. There is a simple recipe, but efficiency matters!

Some of these are conjectural; but given a **complete** guess for suitably large X, one can numerically check the functional equation.

Conductors and bad Euler factors

To "compute" a single hypergeometric L-function, we want the following. (All "recipes" are available in Magma, Sage, and possibly GP/PARI.)

- Gamma factors (i.e., Euler factors at the archimedean place). There is a simple recipe.
- Euler factors and conductor exponents for each **tame** p. There is a simple recipe.
- Euler factors and conductor exponents for each **wild** p. There is a short list of candidates (getting shorter over time...).
- Euler factors for **good** p. We will truncate the Dirichlet series at X^{-s} for some X, which means we need p^a-Frobenius traces for $p^a \leq X$. There is a simple recipe, but efficiency matters!

Some of these are conjectural; but given a **complete** guess for suitably large X, one can numerically check the functional equation.

Conductors and bad Euler factors

To "compute" a single hypergeometric L-function, we want the following. (All "recipes" are available in Magma, Sage, and possibly GP/PARI.)

- Gamma factors (i.e., Euler factors at the archimedean place). There is a simple recipe.
- Euler factors and conductor exponents for each **tame** p. There is a simple recipe.
- Euler factors and conductor exponents for each **wild** p. There is a short list of candidates (getting shorter over time...).
- Euler factors for **good** p. We will truncate the Dirichlet series at X^{-s} for some X, which means we need p^a-Frobenius traces for $p^a \leq X$. There is a simple recipe, but efficiency matters!

Some of these are conjectural; but given a **complete** guess for suitably large X, one can numerically check the functional equation.

Conductors and bad Euler factors

To "compute" a single hypergeometric L-function, we want the following. (All "recipes" are available in Magma, Sage, and possibly GP/PARI.)

- Gamma factors (i.e., Euler factors at the archimedean place). There is a simple recipe.
- Euler factors and conductor exponents for each **tame** p. There is a simple recipe.
- Euler factors and conductor exponents for each **wild** p. There is a short list of candidates (getting shorter over time...).
- Euler factors for **good** p. We will truncate the Dirichlet series at X^{-s} for some X, which means we need p^a-Frobenius traces for $p^a \leq X$. There is a simple recipe, but efficiency matters!

Some of these are conjectural; but given a **complete** guess for suitably large X, one can numerically check the functional equation.

Conductors and bad Euler factors

To "compute" a single hypergeometric L-function, we want the following. (All "recipes" are available in Magma, Sage, and possibly GP/PARI.)

- Gamma factors (i.e., Euler factors at the archimedean place). There is a simple recipe.
- Euler factors and conductor exponents for each **tame** p. There is a simple recipe.
- Euler factors and conductor exponents for each **wild** p. There is a short list of candidates (getting shorter over time...).
- Euler factors for **good** p. We will truncate the Dirichlet series at X^{-s} for some X, which means we need p^a-Frobenius traces for $p^a \leq X$. There is a simple recipe, but efficiency matters!

Some of these are conjectural; but given a **complete** guess for suitably large X, one can numerically check the functional equation.

Frobenius structure

For fixed α, β and p not wild, one can give a **uniform** description of the action of Frob_p on $M_z^{\alpha,\beta}$ in terms of a p-adic analytic **Frobenius structure** on the hypergeometric differential equation (Dwork).

With Grubb we have implemented this in Sage; it works but in practice seems not competitive with the trace formula (next section).

That said, it should be possible to use Frobenius structures to give a new proof of the trace formula (possibly via the comparison between crystalline and Dwork cohomologies). This might to some generalizations to other families (e.g., A-hypergeometric systems) or some further variants (e.g., a q-analogue) which seem less accessible via the current (somewhat indirect) proof.

Frobenius structure

For fixed α, β and p not wild, one can give a **uniform** description of the action of Frob_p on $M_z^{\alpha,\beta}$ in terms of a p-adic analytic **Frobenius structure** on the hypergeometric differential equation (Dwork).

With Grubb we have implemented this in Sage; it works but in practice seems not competitive with the trace formula (next section).

That said, it should be possible to use Frobenius structures to give a new proof of the trace formula (possibly via the comparison between crystalline and Dwork cohomologies). This might to some generalizations to other families (e.g., A-hypergeometric systems) or some further variants (e.g., a q-analogue) which seem less accessible via the current (somewhat indirect) proof.

Frobenius structure

For fixed α, β and p not wild, one can give a **uniform** description of the action of Frob_p on $M_z^{\alpha,\beta}$ in terms of a p-adic analytic **Frobenius structure** on the hypergeometric differential equation (Dwork).

With Grubb we have implemented this in Sage; it works but in practice seems not competitive with the trace formula (next section).

That said, it should be possible to use Frobenius structures to give a new proof of the trace formula (possibly via the comparison between crystalline and Dwork cohomologies). This might to some generalizations to other families (e.g., A-hypergeometric systems) or some further variants (e.g., a q-analogue) which seem less accessible via the current (somewhat indirect) proof.

Contents

Trace formula

For q a power of a good prime p, let $H_q\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\middle| z\right)$ be the trace of Frob_q on $M_z^{\alpha,\beta}$. From work of Greene, Katz, Beukers–Cohen–Mellit, Cohen–Rodriguez Villegas–Watkins, etc., we extract the formula:

$$H_q\left(\begin{matrix}\alpha\\ \beta\end{matrix}\middle| z\right) = \frac{1}{1-q}\sum_{m=0}^{q-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}q^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)_m^*}{(\beta_j)_m^*}\right)[z]^m.$$

Here:

- η_m, ξ_m, D denote some combinatorial quantities (see below);
- $(x)_m^*$ is a p-adic analogue of the Pochhammer symbol (see below);
- $[z] \in \mathbb{Q}_p^{\mathrm{unr}}$ is the multiplicative lift[§] of z.

For fixed q, all of this is very easy to compute efficiently.

[§]Proposed replacement terminology for the historical term “Teichmüller lift”.

Trace formula

For q a power of a good prime p, let $H_q \left({\alpha \atop \beta} \middle| z \right)$ be the trace of Frob_q on $M_z^{\alpha,\beta}$. From work of Greene, Katz, Beukers–Cohen–Mellit, Cohen–Rodriguez Villegas–Watkins, etc., we extract the formula:

$$H_q \left({\alpha \atop \beta} \middle| z \right) = \frac{1}{1-q} \sum_{m=0}^{q-2} (-p)^{\eta_m(\alpha)-\eta_m(\beta)} q^{D+\xi_m(\beta)} \left(\prod_{j=1}^{n} \frac{(\alpha_j)_m^*}{(\beta_j)_m^*} \right) [z]^m.$$

Here:

- η_m, ξ_m, D denote some combinatorial quantities (see below);
- $(x)_m^*$ is a p-adic analogue of the Pochhammer symbol (see below);
- $[z] \in \mathbb{Q}_p^{\mathrm{unr}}$ is the multiplicative lift§ of z.

For fixed q, all of this is very easy to compute efficiently.

§Proposed replacement terminology for the historical term "Teichmüller lift".

Trace formula

For q a power of a good prime p, let $H_q\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\middle| z\right)$ be the trace of Frob_q on $M_z^{\alpha,\beta}$. From work of Greene, Katz, Beukers–Cohen–Mellit, Cohen–Rodriguez Villegas–Watkins, etc., we extract the formula:

$$H_q\left(\begin{matrix}\alpha\\ \beta\end{matrix}\middle| z\right) = \frac{1}{1-q}\sum_{m=0}^{q-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}q^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)_m^*}{(\beta_j)_m^*}\right)[z]^m.$$

Here:

- η_m, ξ_m, D denote some combinatorial quantities (see below);
- $(x)_m^*$ is a p-adic analogue of the Pochhammer symbol (see below);
- $[z] \in \mathbb{Q}_p^{\mathrm{unr}}$ is the multiplicative lift[§] of z.

For fixed q, all of this is very easy to compute efficiently.

[§]Proposed replacement terminology for the historical term "Teichmüller lift".

Trace formula

For q a power of a good prime p, let $H_q\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\middle| z\right)$ be the trace of Frob_q on $M_z^{\alpha,\beta}$. From work of Greene, Katz, Beukers–Cohen–Mellit, Cohen–Rodriguez Villegas–Watkins, etc., we extract the formula:

$$H_q\left(\begin{matrix}\alpha\\ \beta\end{matrix}\middle| z\right) = \frac{1}{1-q}\sum_{m=0}^{q-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}q^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)_m^*}{(\beta_j)_m^*}\right)[z]^m.$$

Here:

- η_m, ξ_m, D denote some combinatorial quantities (see below);
- $(x)_m^*$ is a p-adic analogue of the Pochhammer symbol (see below);
- $[z] \in \mathbb{Q}_p^{\mathrm{unr}}$ is the multiplicative lift[§] of z.

For fixed q, all of this is very easy to compute efficiently.

[§]Proposed replacement terminology for the historical term "Teichmüller lift".

Combinatorial quantities in the trace formula

In the formula

$$H_q\left(\begin{matrix}\alpha\\ \beta\end{matrix}\middle| z\right) := \frac{1}{1-q}\sum_{m=0}^{q-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}q^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)_m^*}{(\beta_j)_m^*}\right)[z]^m$$

the powers of $-p$ and $q = p^f$ are expressed in terms of the following:[¶]

$$\eta_m(x_1,\dots,x_n) := \sum_{j=1}^{n}\sum_{v=0}^{f-1}\left\{p^v\left(x_j+\tfrac{m}{1-q}\right)\right\}-\{p^v x_j\},\ \{x\} := x-\lfloor x\rfloor;$$

$$\xi_m(\beta) := \#\{j : \beta_j = 0\} - \#\left\{j : \beta_j + \tfrac{m}{1-q} = 0\right\};$$

$$D := \frac{w+1-\#\{j : \beta_j = 0\}}{2}.$$

In particular, if we break up $[0,1)$ at the values in $\alpha\cup\beta$, then the powers of $-p$ and q remain constant as $\frac{m}{q-1}$ varies within a subinterval.

[¶] This assumes $0 \notin \alpha$. Otherwise, swap $\alpha \leftrightarrow \beta$ and $z \leftrightarrow 1-z$.

Pochhammer symbols in the trace formula

In the formula

$$H_q\left(\begin{matrix}\alpha\\ \beta\end{matrix}\middle| z\right) = \frac{1}{1-q}\sum_{m=0}^{q-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}q^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)^*_m}{(\beta_j)^*_m}\right)[z]^m$$

the analogue of the Pochhammer symbol is given by

$$(x)^*_m := \frac{\Gamma^*_q\left(x+\frac{m}{1-q}\right)}{\Gamma^*_q(x)}, \qquad \Gamma^*_q(x) := \prod_{v=0}^{f-1}\Gamma_p(\{p^v x\})$$

where $\Gamma_p\colon \mathbb{Z}_p \to \mathbb{Z}_p^\times$ is the Morita p-adic Gamma function. In particular, Γ_p is continuous, $\Gamma_p(0) = 1$, and

$$\Gamma_p(x+1) = \begin{cases} -x\Gamma_p(x) & x \notin p\mathbb{Z}_p \\ -\Gamma_p(x) & x \in p\mathbb{Z}_p. \end{cases}$$

The prime case

Let us now focus on the case $q = p$. In the formula

$$H_p\left(\begin{matrix}\alpha \\ \beta\end{matrix}\middle| z\right) := \frac{1}{1-p}\sum_{m=0}^{p-2} (-p)^{\eta_m(\alpha)-\eta_m(\beta)} p^{D+\xi_m(\beta)} \left(\prod_{j=1}^{n} \frac{(\alpha_j)^*_m}{(\beta_j)^*_m}\right) [z]^m,$$

if we restrict to summands where $\frac{m}{p-1}$ lies between two consecutive values in $\alpha \cup \beta$, then this looks like a truncated hypergeometric series.

Remember that we need to compute this for all good $p \leq X$. If we did this individually, each sum would be over $p-1$ terms, so this would cost roughly $O(X^2)$ time; however, there is clearly a great deal of redundancy. Our goal will be to leverage this redundancy to get this down to $O(X^{1+\epsilon})$.

Note that this still leaves $O(X^{3/2})$ work to deal with higher powers. It may be possible to use a similar approach to reduce this exponent also.

The prime case

Let us now focus on the case $q = p$. In the formula

$$H_p\left(\begin{matrix}\alpha \\ \beta\end{matrix}\middle| z\right) := \frac{1}{1-p}\sum_{m=0}^{p-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}p^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)_m^*}{(\beta_j)_m^*}\right)[z]^m,$$

if we restrict to summands where $\frac{m}{p-1}$ lies between two consecutive values in $\alpha \cup \beta$, then this looks like a truncated hypergeometric series.

Remember that we need to compute this for all good $p \leq X$. If we did this individually, each sum would be over $p-1$ terms, so this would cost roughly $O(X^2)$ time; however, there is clearly a great deal of redundancy. Our goal will be to leverage this redundancy to get this down to $O(X^{1+\epsilon})$.

Note that this still leaves $O(X^{3/2})$ work to deal with higher powers. It may be possible to use a similar approach to reduce this exponent also.

The prime case

Let us now focus on the case $q = p$. In the formula

$$H_p\left(\begin{matrix}\alpha \\ \beta\end{matrix}\middle| z\right) := \frac{1}{1-p}\sum_{m=0}^{p-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}p^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)^*_m}{(\beta_j)^*_m}\right)[z]^m,$$

if we restrict to summands where $\frac{m}{p-1}$ lies between two consecutive values in $\alpha \cup \beta$, then this looks like a truncated hypergeometric series.

Remember that we need to compute this for all good $p \leq X$. If we did this individually, each sum would be over $p-1$ terms, so this would cost roughly $O(X^2)$ time; however, there is clearly a great deal of redundancy. Our goal will be to leverage this redundancy to get this down to $O(X^{1+\epsilon})$.

Note that this still leaves $O(X^{3/2})$ work to deal with higher powers. It may be possible to use a similar approach to reduce this exponent also.

Contents

A minimal example: Wilson primes

The Alhazen–Wilson theorem says that for every prime p, $(p-1)! \equiv -1 \pmod{p}$. A **Wilson prime** is a prime for which $(p-1)! \equiv -1 \pmod{p^2}$. The only known examples are $p = 5, 13, 563$.

Costa–Gerbicz–Harvey computed the reduction of $(p-1)! + 1 \bmod p^2$ for all $p \leq X$ with $X = 2 \times 10^{13}$, using a novel technique to reduce the complexity from $O(X^{2+\epsilon})$ to $O(X^{1+\epsilon})$. Harvey–Sutherland described this in terms of **accumulating remainder trees**, loosely inspired by the structure of the **fast Fourier transform** (FFT) algorithm.

To a first approximation, the idea is to replace the separate computation of $(p-1)! + 1 \pmod{p^2}$ with the serial computation of

$$n! \pmod{\prod_{n<p\leq X} p^2} \qquad \text{for } n = 0, \dots, X-1$$

to eliminate redudancy. However, this must be balanced against making the moduli so large that they slow down the computation.

A minimal example: Wilson primes

The Alhazen–Wilson theorem says that for every prime p, $(p-1)! \equiv -1 \pmod{p}$. A **Wilson prime** is a prime for which $(p-1)! \equiv -1 \pmod{p^2}$. The only known examples are $p = 5, 13, 563$.

Costa–Gerbicz–Harvey computed the reduction of $(p-1)! + 1 \bmod p^2$ for all $p \leq X$ with $X = 2 \times 10^{13}$, using a novel technique to reduce the complexity from $O(X^{2+\epsilon})$ to $O(X^{1+\epsilon})$. Harvey–Sutherland described this in terms of **accumulating remainder trees**, loosely inspired by the structure of the **fast Fourier transform** (FFT) algorithm.

To a first approximation, the idea is to replace the separate computation of $(p-1)! + 1 \pmod{p^2}$ with the serial computation of

$$n! \pmod{\prod_{n<p\leq X} p^2} \qquad \text{for } n = 0, \dots, X-1$$

to eliminate redudancy. However, this must be balanced against making the moduli so large that they slow down the computation.

A minimal example: Wilson primes

The Alhazen–Wilson theorem says that for every prime p, $(p-1)! \equiv -1 \pmod{p}$. A **Wilson prime** is a prime for which $(p-1)! \equiv -1 \pmod{p^2}$. The only known examples are $p = 5, 13, 563$.

Costa–Gerbicz–Harvey computed the reduction of $(p-1)! + 1 \bmod p^2$ for all $p \leq X$ with $X = 2 \times 10^{13}$, using a novel technique to reduce the complexity from $O(X^{2+\epsilon})$ to $O(X^{1+\epsilon})$. Harvey–Sutherland described this in terms of **accumulating remainder trees**, loosely inspired by the structure of the **fast Fourier transform** (FFT) algorithm.

To a first approximation, the idea is to replace the separate computation of $(p-1)! + 1 \pmod{p^2}$ with the serial computation of

$$n! \pmod{\prod_{n<p\leq X} p^2} \qquad \text{for } n = 0, \dots, X-1$$

to eliminate redudancy. However, this must be balanced against making the moduli so large that they slow down the computation.

Accumulating remainder trees

Say we are given integers (or matrices) $A_0, \dots, A_{b-1}$ and integers $m_1, \dots, m_{b-1}$, and we want to compute simultaneously

$$C_j := A_0 \cdots A_{j-1} \pmod{m_j} \qquad (j = 0, \dots, b-1).$$

To simplify, assume $b = 2^\ell$. Form a complete binary tree of depth ℓ with nodes (i,j) where $i = 0, \dots, \ell$ and $j = 0, \dots, 2^{i-1}$. By computing from the leaves to the root, we can compute products over dyadic ranges:

$$\begin{aligned} m_{i,j} &:= m_{j2^{\ell-i}} \cdots m_{(j+1)2^{\ell-i}-1}, \\ A_{i,j} &:= A_{j2^{\ell-i}} \cdots A_{(j+1)2^{\ell-i}-1}. \end{aligned}$$

Then from the root to the leaves, we compute the products $C_{i,j} := A_{i,0} \cdots A_{i,j-1} \pmod{m_{i,j}}$ by writing

$$C_{i,j} = \begin{cases} C_{i-1,\lfloor j/2 \rfloor} \pmod{m_{i,j}} & j \equiv 0 \pmod 2 \\ C_{i-1,\lfloor j/2 \rfloor} A_{i,j-1} \pmod{m_{i,j}} & j \equiv 1 \pmod 2. \end{cases}$$

Accumulating remainder trees

Say we are given integers (or matrices) $A_0, \dots, A_{b-1}$ and integers $m_1, \dots, m_{b-1}$, and we want to compute simultaneously

$$C_j := A_0 \cdots A_{j-1} \pmod{m_j} \qquad (j = 0, \dots, b-1).$$

To simplify, assume $b = 2^\ell$. Form a complete binary tree of depth ℓ with nodes (i,j) where $i = 0, \dots, \ell$ and $j = 0, \dots, 2^{i-1}$. By computing from the leaves to the root, we can compute products over dyadic ranges:

$$m_{i,j} := m_{j2^{\ell-i}} \cdots m_{(j+1)2^{\ell-i}-1},$$
$$A_{i,j} := A_{j2^{\ell-i}} \cdots A_{(j+1)2^{\ell-i}-1}.$$

Then from the root to the leaves, we compute the products $C_{i,j} := A_{i,0} \cdots A_{i,j-1} \pmod{m_{i,j}}$ by writing

$$C_{i,j} = \begin{cases} C_{i-1,\lfloor j/2 \rfloor} \pmod{m_{i,j}} & j \equiv 0 \pmod 2 \\ C_{i-1,\lfloor j/2 \rfloor} A_{i,j-1} \pmod{m_{i,j}} & j \equiv 1 \pmod 2. \end{cases}$$

Accumulating remainder trees

Say we are given integers (or matrices) $A_0, \dots, A_{b-1}$ and integers $m_1, \dots, m_{b-1}$, and we want to compute simultaneously

$$C_j := A_0 \cdots A_{j-1} \pmod{m_j} \qquad (j = 0, \dots, b-1).$$

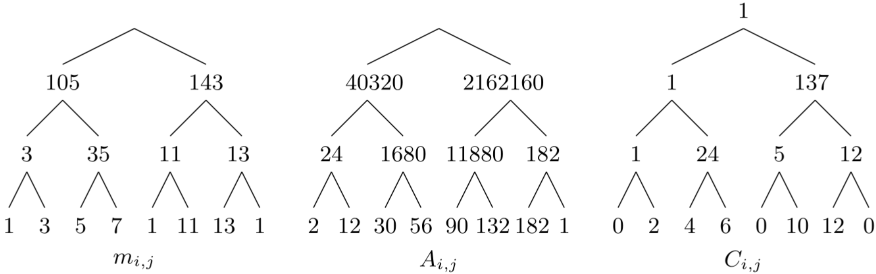
To simplify, assume $b = 2^\ell$. Form a complete binary tree of depth ℓ with nodes (i,j) where $i = 0, \dots, \ell$ and $j = 0, \dots, 2^{i-1}$. By computing from the leaves to the root, we can compute products over dyadic ranges:

$$m_{i,j} := m_{j2^{\ell-i}} \cdots m_{(j+1)2^{\ell-i}-1},$$
$$A_{i,j} := A_{j2^{\ell-i}} \cdots A_{(j+1)2^{\ell-i}-1}.$$

Then from the root to the leaves, we compute the products $C_{i,j} := A_{i,0} \cdots A_{i,j-1} \pmod{m_{i,j}}$ by writing

$$C_{i,j} = \begin{cases} C_{i-1,\lfloor j/2 \rfloor} \pmod{m_{i,j}} & j \equiv 0 \pmod 2 \\ C_{i-1,\lfloor j/2 \rfloor} A_{i,j-1} \pmod{m_{i,j}} & j \equiv 1 \pmod 2. \end{cases}$$

Illustration (Harvey–Sutherland, 2014)



Example: harmonic sums

By forming a product of the matrices $\begin{pmatrix} i^j & 0 \\ 1 & i^j \end{pmatrix}$, for any $\gamma \in \mathbb{Q} \cap (0,1]$ and e, we can efficiently compute for all $p \leq X$ the sums

$$H_{j,\gamma}(p) = \sum_{i=1}^{\lceil \gamma p \rceil - 1} i^{-j} \pmod{p^e} = \sum_{i=1}^{\lceil \gamma p \rceil - 1} \frac{(i!)^j}{((i+1)!)^j} \pmod{p^e}.$$

By applying the functional equation to obtain

$$\log \frac{\Gamma_p(x + \lceil \gamma p \rceil)}{\Gamma_p(\lceil \gamma p \rceil)} = \log \Gamma_p(x) - \sum_{j=1}^{\infty} \frac{(-x)^j}{j} H_{i,\gamma}(j),$$

for any fixed γ we can efficiently compute series expansions of Γ_p around γ modulo p^e for all $p \leq X$.

Example: harmonic sums

By forming a product of the matrices $\begin{pmatrix} i^j & 0 \\ 1 & i^j \end{pmatrix}$, for any $\gamma \in \mathbb{Q} \cap (0,1]$ and e, we can efficiently compute for all $p \leq X$ the sums

$$H_{j,\gamma}(p) = \sum_{i=1}^{\lceil \gamma p \rceil - 1} i^{-j} \pmod{p^e} = \sum_{i=1}^{\lceil \gamma p \rceil - 1} \frac{(i!)^j}{((i+1)!)^j} \pmod{p^e}.$$

By applying the functional equation to obtain

$$\log \frac{\Gamma_p(x + \lceil \gamma p \rceil)}{\Gamma_p(\lceil \gamma p \rceil)} = \log \Gamma_p(x) - \sum_{j=1}^{\infty} \frac{(-x)^j}{j} H_{i,\gamma}(j),$$

for any fixed γ we can efficiently compute series expansions of Γ_p around γ modulo p^e for all $p \leq X$.

Applications in p-adic cohomology

Harvey first observed that the remainder tree technique could be used to speed up computation of L-functions via p-adic cohomology, by exploiting similar redundancies. Further work in this direction has been done by Harvey–Sutherland.

Our application to hypergeometric L-functions is more in the spirit of Costa–Gerbicz–Harvey: we amortize the computation of the trace formula modulo p^e for all $p \leq X$ by exploiting the similarity to a truncated hypergeometric sum. For $e = 1$, this will look very similar to the algorithm for harmonic sums.

Applications in p-adic cohomology

Harvey first observed that the remainder tree technique could be used to speed up computation of L-functions via p-adic cohomology, by exploiting similar redundancies. Further work in this direction has been done by Harvey–Sutherland.

Our application to hypergeometric L-functions is more in the spirit of Costa–Gerbicz–Harvey: we amortize the computation of the trace formula modulo p^e for all $p \leq X$ by exploiting the similarity to a truncated hypergeometric sum. For $e = 1$, this will look very similar to the algorithm for harmonic sums.

Contents

Breaking the trace formula into ranges

Returning to the hypergeometric trace formula with $q = p$:

$$H_p\left(\begin{matrix}\alpha\\ \beta\end{matrix}\middle| z\right) = \frac{1}{1-p}\sum_{m=0}^{p-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}p^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)^*_m}{(\beta_j)^*_m}\right)[z]^m,$$

Label the elements of $\alpha \cup \beta \cup \{0,1\}$ as $0 = \gamma_0 < \cdots < \gamma_s = 1$; set $m_i := \lfloor \gamma_i(p-1)\rfloor$; and focus on the sum over $m \in [m_i, m_{i+1})$ for some i. As noted earlier, there are integers σ_i, τ_i such that

$$(-p)^{\eta_m(\alpha)-\eta_m(\beta)}p^{D+\xi_m(\beta)} = \begin{cases}\tau_i & m = m_i\\ \sigma_i & m_i < m < m_{i+1}.\end{cases}$$

We can thus fix i and focus on computing, for all $p \leq X$,

$$\sum_{m=m_i+1}^{m_{i+1}-1}\left(\prod_{j=1}^{n}\frac{(\alpha_j)^*_m}{(\beta_j)^*_m}\right)[z]^m.$$

Breaking the trace formula into ranges

Returning to the hypergeometric trace formula with $q = p$:

$$H_p\left(\begin{matrix}\alpha \\ \beta\end{matrix}\middle| z\right) = \frac{1}{1-p}\sum_{m=0}^{p-2}(-p)^{\eta_m(\alpha)-\eta_m(\beta)}p^{D+\xi_m(\beta)}\left(\prod_{j=1}^{n}\frac{(\alpha_j)_m^*}{(\beta_j)_m^*}\right)[z]^m,$$

Label the elements of $\alpha \cup \beta \cup \{0,1\}$ as $0 = \gamma_0 < \cdots < \gamma_s = 1$; set $m_i := \lfloor \gamma_i(p-1) \rfloor$; and focus on the sum over $m \in [m_i, m_{i+1})$ for some i. As noted earlier, there are integers σ_i, τ_i such that

$$(-p)^{\eta_m(\alpha)-\eta_m(\beta)}p^{D+\xi_m(\beta)} = \begin{cases}\tau_i & m = m_i \\ \sigma_i & m_i < m < m_{i+1}.\end{cases}$$

We can thus fix i and focus on computing, for all $p \leq X$,

$$\sum_{m=m_i+1}^{m_{i+1}-1}\left(\prod_{j=1}^{n}\frac{(\alpha_j)_m^*}{(\beta_j)_m^*}\right)[z]^m.$$

Change of endpoints

We need to shift indices so that the sums all run from 1. That is, we want to take $m = m_i + k$ and sum over $k = 1, \dots, m_{i+1} - m_i - 1$.

Write $\gamma_i = \frac{a_i}{b_i}$ in lowest terms, fix $c \in (\mathbb{Z}/b_i\mathbb{Z})^\times$, and restrict attention to $p \equiv c \pmod{b_i}$. We then have

$$m_i = \gamma_i(p-1) - \gamma_{i,c} \text{ where } a_i(p-1) = m_i b_i + r_i, \gamma_{i,c} = \frac{r_i}{b_i} \in [0,1).$$

For $\gamma \in \alpha \cup \beta$, $(\gamma)^*_m = \Gamma_p(\{\gamma + \frac{m}{1-p}\})/\Gamma_p(\gamma)$ and

$$\left\{\gamma + \frac{m}{1-p}\right\} = k + (k - \gamma_{i,c})\tfrac{p}{1-p} + h_c(\gamma, \gamma_i)$$

where

$$h_c(\gamma,\gamma_i) := \gamma - \gamma_i + \iota(\gamma,\gamma_i) - \gamma_{i,c} \in (-1,1], \quad \iota(x,y) := \begin{cases} 1 & x \le y \\ 0 & x > y. \end{cases}$$

Change of endpoints

We need to shift indices so that the sums all run from 1. That is, we want to take $m = m_i + k$ and sum over $k = 1, \ldots, m_{i+1} - m_i - 1$.

Write $\gamma_i = \frac{a_i}{b_i}$ in lowest terms, fix $c \in (\mathbb{Z}/b_i\mathbb{Z})^\times$, and restrict attention to $p \equiv c \pmod{b_i}$. We then have

$$m_i = \gamma_i(p-1) - \gamma_{i,c} \text{ where } a_i(p-1) = m_i b_i + r_i, \gamma_{i,c} = \frac{r_i}{b_i} \in [0,1).$$

For $\gamma \in \alpha \cup \beta$, $(\gamma)^*_m = \Gamma_p(\{\gamma + \frac{m}{1-p}\})/\Gamma_p(\gamma)$ and

$$\left\{\gamma + \frac{m}{1-p}\right\} = k + (k - \gamma_{i,c})\tfrac{p}{1-p} + h_c(\gamma, \gamma_i)$$

where

$$h_c(\gamma,\gamma_i) := \gamma - \gamma_i + \iota(\gamma,\gamma_i) - \gamma_{i,c} \in (-1,1], \quad \iota(x,y) := \begin{cases} 1 & x \le y \\ 0 & x > y. \end{cases}$$

Change of endpoints

We need to shift indices so that the sums all run from 1. That is, we want to take $m = m_i + k$ and sum over $k = 1, \ldots, m_{i+1} - m_i - 1$.

Write $\gamma_i = \frac{a_i}{b_i}$ in lowest terms, fix $c \in (\mathbb{Z}/b_i\mathbb{Z})^\times$, and restrict attention to $p \equiv c \pmod{b_i}$. We then have

$$m_i = \gamma_i(p-1) - \gamma_{i,c} \text{ where } a_i(p-1) = m_i b_i + r_i, \gamma_{i,c} = \frac{r_i}{b_i} \in [0,1).$$

For $\gamma \in \alpha \cup \beta$, $(\gamma)^*_m = \Gamma_p(\{\gamma + \frac{m}{1-p}\})/\Gamma_p(\gamma)$ and

$$\left\{\gamma + \frac{m}{1-p}\right\} = k + (k - \gamma_{i,c})\tfrac{p}{1-p} + h_c(\gamma, \gamma_i)$$

where

$$h_c(\gamma,\gamma_i) := \gamma - \gamma_i + \iota(\gamma,\gamma_i) - \gamma_{i,c} \in (-1,1], \quad \iota(x,y) := \begin{cases} 1 & x \leq y \\ 0 & x > y. \end{cases}$$

The situation mod p

Recall that we need to sum for all $p \leq X$,

$$\sum_{m=m_i+1}^{m_{i+1}-1} \left(\prod_{j=1}^{n} \frac{(\alpha_j)^*_m}{(\beta_j)^*_m} \right) [z]^m.$$

Say we only want the trace modulo p for each $p \leq X$. Then we are reduced to summing

$$\sum_{k=1}^{m_{i+1}-m_i-1} \prod_{j=0}^{k-1} \frac{z_f f_{i,c}(k)}{z_g g_{i,c}(k)} \pmod{p},$$

where $z = \frac{z_f}{z_g}$ in lowest terms and for some positive integer b,

$$f_{i,c}(k) := b \prod_{j=1}^{n} (h_c(\alpha_j, \gamma_i) + k), \qquad g_{i,c}(k) := b \prod_{j=1}^{n} (h_c(\beta_j, \gamma_i) + k).$$

The situation mod p

Recall that we need to sum for all $p \leq X$,

$$\sum_{m=m_i+1}^{m_{i+1}-1} \left(\prod_{j=1}^{n} \frac{(\alpha_j)_m^*}{(\beta_j)_m^*} \right) [z]^m.$$

Say we only want the trace modulo p for each $p \leq X$. Then we are reduced to summing

$$\sum_{k=1}^{m_{i+1}-m_i-1} \prod_{j=0}^{k-1} \frac{z_f f_{i,c}(k)}{z_g g_{i,c}(k)} \pmod{p},$$

where $z = \frac{z_f}{z_g}$ in lowest terms and for some positive integer b,

$$f_{i,c}(k) := b \prod_{j=1}^{n} (h_c(\alpha_j, \gamma_i) + k), \qquad g_{i,c}(k) := b \prod_{j=1}^{n} (h_c(\beta_j, \gamma_i) + k).$$

The situation mod p (continued)

Using a remainder tree, we can compute products of matrices of the form

$$A_{i,c}(k) := \begin{pmatrix} z_g g_{i,c}(k) & 0 \\ z_g g_{i,c}(k) & z_f f_{i,c}(k) \end{pmatrix}.$$

For

$$S_i(p) := A_{i,c}(1) \cdots A_{i,c}(m_{i+1} - m_i - 1),$$

we have

$$\frac{S_i(p)_{21}}{S_i(p)_{11}} \equiv \sum_{k=1}^{m_{i+1}-m_i-1} \prod_{j=0}^{k-1} \frac{z_f f_{i,c}(k)}{z_g g_{i,c}(k)} \equiv \sum_{m=m_i+1}^{m_{i+1}-1} \left(\prod_{j=1}^{n} \frac{(\alpha_j)^*_m}{(\beta_j)^*_m} \right) [z]^m \pmod{p}.$$

This is **extremely** fast in practice (see our paper from ANTS XIV, 2020).

The situation mod p (continued)

Using a remainder tree, we can compute products of matrices of the form

$$A_{i,c}(k) := \begin{pmatrix} z_g g_{i,c}(k) & 0 \\ z_g g_{i,c}(k) & z_f f_{i,c}(k) \end{pmatrix}.$$

For

$$S_i(p) := A_{i,c}(1) \cdots A_{i,c}(m_{i+1} - m_i - 1),$$

we have

$$\frac{S_i(p)_{21}}{S_i(p)_{11}} \equiv \sum_{k=1}^{m_{i+1}-m_i-1} \prod_{j=0}^{k-1} \frac{z_f f_{i,c}(k)}{z_g g_{i,c}(k)} \equiv \sum_{m=m_i+1}^{m_{i+1}-1} \left(\prod_{j=1}^{n} \frac{(\alpha_j)^*_m}{(\beta_j)^*_m} \right) [z]^m \pmod{p}.$$

This is **extremely** fast in practice (see our paper from ANTS XIV, 2020).

The situation mod p (continued)

Using a remainder tree, we can compute products of matrices of the form

$$A_{i,c}(k) := \begin{pmatrix} z_g g_{i,c}(k) & 0 \\ z_g g_{i,c}(k) & z_f f_{i,c}(k) \end{pmatrix}.$$

For

$$S_i(p) := A_{i,c}(1) \cdots A_{i,c}(m_{i+1} - m_i - 1),$$

we have

$$\frac{S_i(p)_{21}}{S_i(p)_{11}} \equiv \sum_{k=1}^{m_{i+1}-m_i-1} \prod_{j=0}^{k-1} \frac{z_f f_{i,c}(k)}{z_g g_{i,c}(k)} \equiv \sum_{m=m_i+1}^{m_{i+1}-1} \left(\prod_{j=1}^{n} \frac{(\alpha_j)_m^*}{(\beta_j)_m^*} \right) [z]^m \pmod{p}.$$

This is **extremely** fast in practice (see our paper from ANTS XIV, 2020).

Contents

Some complications

In the general case, it is sufficient to compute modulo p^e for $e = \lfloor (w+1)/2 \rfloor$ where w is the motivic weight (at least for $p > 4n^2$). There are several additional complications to be overcome.

- We cannot ignore the difference between $[z]$ and z. It is easy to compute $[z]$ for any given p, but it does not behave uniformly.
- We need to incorporate the expansion of Γ_p around some rational arguments (which we already know how to compute in average polynomial time).
- The functional equation relates $\Gamma_p(x)$ to $\Gamma_p(x+1)$, not $\Gamma_p(x + \frac{1}{1-p})$.

The solution we describe here will be presented at ANTS XVI in July 2024.

Some complications

In the general case, it is sufficient to compute modulo p^e for $e = \lfloor (w+1)/2 \rfloor$ where w is the motivic weight (at least for $p > 4n^2$). There are several additional complications to be overcome.

- We cannot ignore the difference between $[z]$ and z. It is easy to compute $[z]$ for any given p, but it does not behave uniformly.
- We need to incorporate the expansion of Γ_p around some rational arguments (which we already know how to compute in average polynomial time).
- The functional equation relates $\Gamma_p(x)$ to $\Gamma_p(x+1)$, not $\Gamma_p(x + \frac{1}{1-p})$.

The solution we describe here will be presented at ANTS XVI in July 2024.

Some complications

In the general case, it is sufficient to compute modulo p^e for $e = \lfloor (w+1)/2 \rfloor$ where w is the motivic weight (at least for $p > 4n^2$). There are several additional complications to be overcome.

- We cannot ignore the difference between $[z]$ and z. It is easy to compute $[z]$ for any given p, but it does not behave uniformly.
- We need to incorporate the expansion of Γ_p around some rational arguments (which we already know how to compute in average polynomial time).
- The functional equation relates $\Gamma_p(x)$ to $\Gamma_p(x+1)$, not $\Gamma_p(x + \frac{1}{1-p})$.

The solution we describe here will be presented at ANTS XVI in July 2024.

Some complications

In the general case, it is sufficient to compute modulo p^e for $e = \lfloor (w+1)/2 \rfloor$ where w is the motivic weight (at least for $p > 4n^2$). There are several additional complications to be overcome.

- We cannot ignore the difference between $[z]$ and z. It is easy to compute $[z]$ for any given p, but it does not behave uniformly.
- We need to incorporate the expansion of Γ_p around some rational arguments (which we already know how to compute in average polynomial time).
- The functional equation relates $\Gamma_p(x)$ to $\Gamma_p(x+1)$, not $\Gamma_p(x + \frac{1}{1-p})$.

The solution we describe here will be presented at ANTS XVI in July 2024.

Some complications

In the general case, it is sufficient to compute modulo p^e for $e = \lfloor (w+1)/2 \rfloor$ where w is the motivic weight (at least for $p > 4n^2$). There are several additional complications to be overcome.

- We cannot ignore the difference between $[z]$ and z. It is easy to compute $[z]$ for any given p, but it does not behave uniformly.
- We need to incorporate the expansion of Γ_p around some rational arguments (which we already know how to compute in average polynomial time).
- The functional equation relates $\Gamma_p(x)$ to $\Gamma_p(x+1)$, not $\Gamma_p(x + \frac{1}{1-p})$.

The solution we describe here will be presented at ANTS XVI in July 2024.

Harvey's generic prime construction

A key idea comes from the work of Harvey: consider products of matrices over $\mathbb{Z}[x]/(x^e)$ instead of $\mathbb{Z}$. Then for each prime p, we can take the result and replace x with something divisible by p which does **not** need to be computed by a matrix product.

For example, if the only issue were the discrepancy between z and $[z]$, we could replace $[z]$ with $z(1+x)$ and then afterwards substitute $x \mapsto [z]/z - 1$, which we can compute efficiently for individual p. (In Harvey's setting he needs to substitute $x \mapsto p$.)

In practice, we instead replace $\mathbb{Z}$ with the **noncommutative** ring of lower triangular $e \times e$ matrices over $\mathbb{Z}$. This contains $\mathbb{Z}[x]/(x^e)$ (as banded matrices) but allows for additional operations, crucially including $x \mapsto cx$.

Harvey's generic prime construction

A key idea comes from the work of Harvey: consider products of matrices over $\mathbb{Z}[x]/(x^e)$ instead of $\mathbb{Z}$. Then for each prime p, we can take the result and replace x with something divisible by p which does **not** need to be computed by a matrix product.

For example, if the only issue were the discrepancy between z and $[z]$, we could replace $[z]$ with $z(1+x)$ and then afterwards substitute $x \mapsto [z]/z - 1$, which we can compute efficiently for individual p. (In Harvey's setting he needs to substitute $x \mapsto p$.)

In practice, we instead replace $\mathbb{Z}$ with the **noncommutative** ring of lower triangular $e \times e$ matrices over $\mathbb{Z}$. This contains $\mathbb{Z}[x]/(x^e)$ (as banded matrices) but allows for additional operations, crucially including $x \mapsto cx$.

Harvey's generic prime construction

A key idea comes from the work of Harvey: consider products of matrices over $\mathbb{Z}[x]/(x^e)$ instead of $\mathbb{Z}$. Then for each prime p, we can take the result and replace x with something divisible by p which does **not** need to be computed by a matrix product.

For example, if the only issue were the discrepancy between z and $[z]$, we could replace $[z]$ with $z(1+x)$ and then afterwards substitute $x \mapsto [z]/z - 1$, which we can compute efficiently for individual p. (In Harvey's setting he needs to substitute $x \mapsto p$.)

In practice, we instead replace $\mathbb{Z}$ with the **noncommutative** ring of lower triangular $e \times e$ matrices over $\mathbb{Z}$. This contains $\mathbb{Z}[x]/(x^e)$ (as banded matrices) but allows for additional operations, crucially including $x \mapsto cx$.

Factorization of the quotient

The ratio of the k-th term in our sum to the 1st term can be interpreted as

$$[z]^{k-1} \prod_{\gamma \in \beta}^{\gamma \in \alpha} \frac{\Gamma_p\left(h_c(\gamma, \gamma_i) + k + \frac{(k-\gamma_{i,c})p}{1-p}\right)}{\Gamma_p\left(h_c(\gamma, \gamma_i) + 1 + \frac{(1-\gamma_{i,c})p}{1-p}\right)}$$

where $\prod_{\gamma \in \beta}^{\gamma \in \alpha}$ means take the product over $\gamma = \alpha_1, \dots, \alpha_n$ divided by the product over $\gamma = \beta_1, \dots, \beta_n$.

Define the power series

$$R_i(x) := \prod_{\gamma \in \beta}^{\gamma \in \alpha} \frac{\Gamma_p(x + h_c(\gamma, \gamma_i) + 1)}{\Gamma_p(h_c(\gamma, \gamma_i) + 1)}.$$

We can then write the above ratio as

$$\left(\frac{[z]}{z}\right)^{k-1} \frac{R_i((k-\gamma_{i,c})\frac{p}{1-p})}{R_i((1-\gamma_{i,c})\frac{p}{1-p})} \cdot \prod_{j=1}^{k-1} \frac{f_{i,c}(x+j)}{g_{i,c}(x+j)}\Bigg|_{x=(k-\gamma_{i,c})\frac{p}{1-p}}.$$

Factorization of the quotient

The ratio of the k-th term in our sum to the 1st term can be interpreted as

$$[z]^{k-1} \prod_{\gamma \in \beta}^{\gamma \in \alpha} \frac{\Gamma_p\left(h_c(\gamma, \gamma_i) + k + \frac{(k-\gamma_{i,c})p}{1-p}\right)}{\Gamma_p\left(h_c(\gamma, \gamma_i) + 1 + \frac{(1-\gamma_{i,c})p}{1-p}\right)}$$

where $\prod_{\gamma \in \beta}^{\gamma \in \alpha}$ means take the product over $\gamma = \alpha_1, \dots, \alpha_n$ divided by the product over $\gamma = \beta_1, \dots, \beta_n$.

Define the power series

$$R_i(x) := \prod_{\gamma \in \beta}^{\gamma \in \alpha} \frac{\Gamma_p(x + h_c(\gamma, \gamma_i) + 1)}{\Gamma_p(h_c(\gamma, \gamma_i) + 1)}.$$

We can then write the above ratio as

$$\left(\frac{[z]}{z}\right)^{k-1} \frac{R_i((k-\gamma_{i,c})\frac{p}{1-p})}{R_i((1-\gamma_{i,c})\frac{p}{1-p})} \cdot \prod_{j=1}^{k-1} \frac{f_{i,c}(x+j)}{g_{i,c}(x+j)}\Bigg|_{x=(k-\gamma_{i,c})\frac{p}{1-p}}.$$

Factorization of the quotient

The ratio of the k-th term in our sum to the 1st term can be interpreted as

$$[z]^{k-1} \prod_{\gamma \in \beta}^{\gamma \in \alpha} \frac{\Gamma_p\left(h_c(\gamma, \gamma_i) + k + \frac{(k-\gamma_{i,c})p}{1-p}\right)}{\Gamma_p\left(h_c(\gamma, \gamma_i) + 1 + \frac{(1-\gamma_{i,c})p}{1-p}\right)}$$

where $\prod_{\gamma \in \beta}^{\gamma \in \alpha}$ means take the product over $\gamma = \alpha_1, \dots, \alpha_n$ divided by the product over $\gamma = \beta_1, \dots, \beta_n$.

Define the power series

$$R_i(x) := \prod_{\gamma \in \beta}^{\gamma \in \alpha} \frac{\Gamma_p(x + h_c(\gamma, \gamma_i) + 1)}{\Gamma_p(h_c(\gamma, \gamma_i) + 1)}.$$

We can then write the above ratio as

$$\left(\frac{[z]}{z}\right)^{k-1} \frac{R_i((k-\gamma_{i,c})\frac{p}{1-p})}{R_i((1-\gamma_{i,c})\frac{p}{1-p})} \cdot \prod_{j=1}^{k-1} \frac{f_{i,c}(x+j)}{g_{i,c}(x+j)}\Bigg|_{x=(k-\gamma_{i,c})\frac{p}{1-p}}.$$

Factorization of the quotient (continued)

In the previous expression, the factor not involving j, namely

$$\left(\frac{[z]}{z}\right)^{k-1} \frac{R_i((k-\gamma_{i,c})\frac{p}{1-p})}{R_i((1-\gamma_{i,c})\frac{p}{1-p})},$$

depends on k in a usefully simple way: it can be written as

$$\sum_{h=0}^{e-1} c_{i,h}(p)\left((k-\gamma_{i,c})\tfrac{p}{1-p}\right)^h \pmod{p^e}$$

for some $c_{i,h}(p)$ independent of k. Conveniently, we do **not** have to worry about how these are computed when forming the matrix product!

Form of the matrix product

We apply remainder trees to multiply block matrices with $e \times e$ blocks:

$$A_{i,c}(k) := (\text{scalar}) \begin{pmatrix} \delta_{h_1,h_2} & 0 \\ (k-\gamma_{i,c})^{e-h_2}\delta_{h_1,h_2} & \left(\frac{f_{i,c}(x+k)}{g_{i,c}(x+k)}\right)^{[h_1-h_2]} \end{pmatrix}$$

where $f(x)^{[h]}$ means the coefficient of x^h in $f(x)$. The effect of adding $A_{i,c}(k)$ to the product is to increment (lower left)/(upper left) by

$$Q_{h_1,h_2}(k) = (k-\gamma_{i,c})^{h_2} \left(\prod_{j=1}^{k-1} \frac{f_{i,c}(x+j)}{g_{i,c}(x+j)} \right)^{[h_2-h_1]}$$

which we combine with the $c_{i,h}(p)$ to get what we want:

$$\sum_k \sum_{h_1,h_2} c_{i,e-h_1} Q_{h_1,h_2}(k) \left(\frac{p}{1-p}\right)^{e-h_2}.$$