Hypergeometric motives

Kiran S. Kedlaya

Department of Mathematics, University of California San Diego
kedlaya@ucsd.edu
These slides can be downloaded from https://kskedlaya.org/slides/.

Atelier PARI/GP 2026
Université de Bordeaux
Jan. 15, 2026

Supported by (grant DMS-2401536) and UC San Diego (Warschawski Professorship).

I acknowledge that my workplace occupies unceded ancestral land of the Kumeyaay Nation.

What is a hypergeometric motive?

D. Roberts and F. Rodriguez Villegas, Hypergeometric motives, *Notices of the AMS* **69** (2022), 914–929.

Executive summary: to certain discrete parameters α, β, one associates a family of smooth projective varieties over $\mathbb{P}^1 \setminus \{0, 1, \infty\}$ with a period integral satisfying the hypergeometric differential equation; this generalizes the classical observation of Gauss for elliptic integrals. For each $t \in \mathbb{Q} \setminus \{0, 1\}$, we can form an associated L-function; this recovers many familiar examples (e.g., elliptic curves, some hyperelliptic curves of low genus, some K3 surfaces) and many exotic ones.

Warning: many "true" statements about HGMs are not well-documented in the literature.

What is a hypergeometric motive?

D. Roberts and F. Rodriguez Villegas, Hypergeometric motives, *Notices of the AMS* **69** (2022), 914–929.

Executive summary: to certain discrete parameters α, β, one associates a family of smooth projective varieties over $\mathbb{P}^1 \setminus \{0, 1, \infty\}$ with a period integral satisfying the hypergeometric differential equation; this generalizes the classical observation of Gauss for elliptic integrals. For each $t \in \mathbb{Q} \setminus \{0, 1\}$, we can form an associated L-function; this recovers many familiar examples (e.g., elliptic curves, some hyperelliptic curves of low genus, some K3 surfaces) and many exotic ones.

Warning: many "true" statements about HGMs are not well-documented in the literature.

What is a hypergeometric motive?

D. Roberts and F. Rodriguez Villegas, Hypergeometric motives, *Notices of the AMS* **69** (2022), 914–929.

Executive summary: to certain discrete parameters α, β, one associates a family of smooth projective varieties over $\mathbb{P}^1 \setminus \{0, 1, \infty\}$ with a period integral satisfying the hypergeometric differential equation; this generalizes the classical observation of Gauss for elliptic integrals. For each $t \in \mathbb{Q} \setminus \{0, 1\}$, we can form an associated L-function; this recovers many familiar examples (e.g., elliptic curves, some hyperelliptic curves of low genus, some K3 surfaces) and many exotic ones.

Warning: many "true" statements about HGMs are not well-documented in the literature.

Parameters for a hypergeometric family

Over $\mathbb{C}$, a hypergeometric family is specified by a positive integer n and two tuples $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ of complex numbers. The family does not change if we replace α, β by $\alpha + c, \beta + c$ for some $c \in \mathbb{C}$ *or* shift any α_i or β_j by an integer.[1] To avoid degenerate cases, we should also assume $\alpha_i \not\equiv \beta_j \pmod{\mathbb{Z}}$ for all i, j.

To get a family definable over $\mathbb{Q}$, we first require $\alpha_i, \beta_j \in \mathbb{Q} \cap [0, 1)$ (and $\alpha_i \neq \beta_j$ for all i, j). We next impose the following condition: the multiplicity of a reduced fraction $\frac{r}{s} \in \mathbb{Q} \cap [0, 1)$ in α or β depends only on s.

The classical Gauss hypergeometric equation corresponds to $\alpha = (\frac{1}{2}, \frac{1}{2}), \beta = (0, 0)$.

[1] It is a classical convention to normalize $\beta_n = 1$, but we won't do that here.

Parameters for a hypergeometric family

Over $\mathbb{C}$, a hypergeometric family is specified by a positive integer n and two tuples $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ of complex numbers. The family does not change if we replace α, β by $\alpha + c, \beta + c$ for some $c \in \mathbb{C}$ *or* shift any α_i or β_j by an integer.[1] To avoid degenerate cases, we should also assume $\alpha_i \not\equiv \beta_j \pmod{\mathbb{Z}}$ for all i, j.

To get a family definable over $\mathbb{Q}$, we first require $\alpha_i, \beta_j \in \mathbb{Q} \cap [0, 1)$ (and $\alpha_i \neq \beta_j$ for all i, j). We next impose the following condition: the multiplicity of a reduced fraction $\frac{r}{s} \in \mathbb{Q} \cap [0, 1)$ in α or β depends only on s.

The classical Gauss hypergeometric equation corresponds to $\alpha = (\frac{1}{2}, \frac{1}{2}), \beta = (0, 0)$.

[1] It is a classical convention to normalize $\beta_n = 1$, but we won't do that here.

Parameters for a hypergeometric family

Over $\mathbb{C}$, a hypergeometric family is specified by a positive integer n and two tuples $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\beta = (\beta_1, \ldots, \beta_n)$ of complex numbers. The family does not change if we replace α, β by $\alpha + c, \beta + c$ for some $c \in \mathbb{C}$ *or* shift any α_i or β_j by an integer.[1] To avoid degenerate cases, we should also assume $\alpha_i \not\equiv \beta_j \pmod{\mathbb{Z}}$ for all i, j.

To get a family definable over $\mathbb{Q}$, we first require $\alpha_i, \beta_j \in \mathbb{Q} \cap [0, 1)$ (and $\alpha_i \neq \beta_j$ for all i, j). We next impose the following condition: the multiplicity of a reduced fraction $\frac{r}{s} \in \mathbb{Q} \cap [0, 1)$ in α or β depends only on s.

The classical Gauss hypergeometric equation corresponds to $\alpha = (\frac{1}{2}, \frac{1}{2}), \beta = (0, 0)$.

[1] It is a classical convention to normalize $\beta_n = 1$, but we won't do that here.

Hypergeometric L-functions

Given α, β as above, for each $t \in \mathbb{Q} \setminus \{0, 1\}$ we get a hypergeometric L-function of degree n:

$$H_{\alpha,\beta,z}(s) = \prod_p L_{\alpha,\beta,z,p}(p^{-s})^{-1}.$$

The motivic weight (and Hodge numbers) are computed by the **zigzag function** of α, β.

The Euler factors of $H_{\alpha,\beta,z}(s)$ can be classified as follows:

- **good**: not tame or wild. Good Euler factors can be computed by a formula of Beukers–Cohen–Mellit, or a p-adic translation by Cohen–Rodriguez Villegas–Watkins.
- **tame**: not wild, but t reduces to one of $0, 1, \infty$ mod p. Tame Euler factors (and conductor exponents) can be computed by a variant of the formula in the good case.
- **wild**: some α_i or β_j is not integral at p. Wild Euler factors (and conductor exponents) are not fully understood, but we have good conjectures in many cases. Moreover, a correct guess for **all** wild Euler factors and conductor exponents can be identified empirically using an approximate functional equation.

Hypergeometric L-functions

Given α, β as above, for each $t \in \mathbb{Q} \setminus \{0, 1\}$ we get a hypergeometric L-function of degree n:

$$H_{\alpha,\beta,z}(s) = \prod_p L_{\alpha,\beta,z,p}(p^{-s})^{-1}.$$

The motivic weight (and Hodge numbers) are computed by the **zigzag function** of α, β.

The Euler factors of $H_{\alpha,\beta,z}(s)$ can be classified as follows:

- **good**: not tame or wild. Good Euler factors can be computed by a formula of Beukers–Cohen–Mellit, or a p-adic translation by Cohen–Rodriguez Villegas–Watkins.
- **tame**: not wild, but t reduces to one of $0, 1, \infty$ mod p. Tame Euler factors (and conductor exponents) can be computed by a variant of the formula in the good case.
- **wild**: some α_i or β_j is not integral at p. Wild Euler factors (and conductor exponents) are not fully understood, but we have good conjectures in many cases. Moreover, a correct guess for **all** wild Euler factors and conductor exponents can be identified empirically using an approximate functional equation.

Hypergeometric L-functions

Given α, β as above, for each $t \in \mathbb{Q} \setminus \{0, 1\}$ we get a hypergeometric L-function of degree n:

$$H_{\alpha,\beta,z}(s) = \prod_p L_{\alpha,\beta,z,p}(p^{-s})^{-1}.$$

The motivic weight (and Hodge numbers) are computed by the **zigzag function** of α, β.

The Euler factors of $H_{\alpha,\beta,z}(s)$ can be classified as follows:

- **good**: not tame or wild. Good Euler factors can be computed by a formula of Beukers–Cohen–Mellit, or a p-adic translation by Cohen–Rodriguez Villegas–Watkins.
- **tame**: not wild, but t reduces to one of $0, 1, \infty$ mod p. Tame Euler factors (and conductor exponents) can be computed by a variant of the formula in the good case.
- **wild**: some α_i or β_j is not integral at p. Wild Euler factors (and conductor exponents) are not fully understood, but we have good conjectures in many cases. Moreover, a correct guess for **all** wild Euler factors and conductor exponents can be identified empirically using an approximate functional equation.

Hypergeometric L-functions

Given α, β as above, for each $t \in \mathbb{Q} \setminus \{0, 1\}$ we get a hypergeometric L-function of degree n:

$$H_{\alpha,\beta,z}(s) = \prod_p L_{\alpha,\beta,z,p}(p^{-s})^{-1}.$$

The motivic weight (and Hodge numbers) are computed by the **zigzag function** of α, β.

The Euler factors of $H_{\alpha,\beta,z}(s)$ can be classified as follows:

- **good**: not tame or wild. Good Euler factors can be computed by a formula of Beukers–Cohen–Mellit, or a p-adic translation by Cohen–Rodriguez Villegas–Watkins.
- **tame**: not wild, but t reduces to one of $0, 1, \infty$ mod p. Tame Euler factors (and conductor exponents) can be computed by a variant of the formula in the good case.
- **wild**: some α_i or β_j is not integral at p. Wild Euler factors (and conductor exponents) are not fully understood, but we have good conjectures in many cases. Moreover, a correct guess for **all** wild Euler factors and conductor exponents can be identified empirically using an approximate functional equation.

Hypergeometric L-functions

Given α, β as above, for each $t \in \mathbb{Q} \setminus \{0, 1\}$ we get a hypergeometric L-function of degree n:

$$H_{\alpha,\beta,z}(s) = \prod_p L_{\alpha,\beta,z,p}(p^{-s})^{-1}.$$

The motivic weight (and Hodge numbers) are computed by the **zigzag function** of α, β.

The Euler factors of $H_{\alpha,\beta,z}(s)$ can be classified as follows:

- **good**: not tame or wild. Good Euler factors can be computed by a formula of Beukers–Cohen–Mellit, or a p-adic translation by Cohen–Rodriguez Villegas–Watkins.
- **tame**: not wild, but t reduces to one of $0, 1, \infty$ mod p. Tame Euler factors (and conductor exponents) can be computed by a variant of the formula in the good case.
- **wild**: some α_i or β_j is not integral at p. Wild Euler factors (and conductor exponents) are not fully understood, but we have good conjectures in many cases. Moreover, a correct guess for **all** wild Euler factors and conductor exponents can be identified empirically using an approximate functional equation.

Implementations

Hypergeometric motives and L-functions are available (to various extents) in PARI/GP (Cohen, Rodriguez Villegas), Magma (Watkins), and SageMath (Chapoton, K); the second and third implementations are ports of their predecessors. **Warning**: the parameter z in SageMath (and most literature) corresponds to $1/z$ in PARI/GP and Magma.

Hypergeometric families and motives are also available in beta LMFDB. Hypergeometric L-functions are not yet included in most cases.

This project (Costa, K, Roe) implements in C/Cython/SageMath an **average polynomial time** algorithm to compute Frobenius traces at p for all good $p \leq X$ in time $\tilde{O}(X)$ (see the link for references). Perhaps this can be ported to PARI?

Implementations

Hypergeometric motives and L-functions are available (to various extents) in PARI/GP (Cohen, Rodriguez Villegas), Magma (Watkins), and SageMath (Chapoton, K); the second and third implementations are ports of their predecessors. **Warning**: the parameter z in SageMath (and most literature) corresponds to $1/z$ in PARI/GP and Magma.

Hypergeometric families and motives are also available in beta LMFDB. Hypergeometric L-functions are not yet included in most cases.

This project (Costa, K, Roe) implements in C/Cython/SageMath an **average polynomial time** algorithm to compute Frobenius traces at p for all good $p \leq X$ in time $\tilde{O}(X)$ (see the link for references). Perhaps this can be ported to PARI?

Implementations

Hypergeometric motives and L-functions are available (to various extents) in PARI/GP (Cohen, Rodriguez Villegas), Magma (Watkins), and SageMath (Chapoton, K); the second and third implementations are ports of their predecessors. **Warning**: the parameter z in SageMath (and most literature) corresponds to $1/z$ in PARI/GP and Magma.

Hypergeometric families and motives are also available in beta LMFDB. Hypergeometric L-functions are not yet included in most cases.

This project (Costa, K, Roe) implements in C/Cython/SageMath an **average polynomial time** algorithm to compute Frobenius traces at p for all good $p \leq X$ in time $\tilde{O}(X)$ (see the link for references). Perhaps this can be ported to PARI?

A model: truncated hypergeometric sums

Computing the coefficient of p^{-s} in a hypergeometric L-function reduces (very roughly) to computing a truncated hypergeometric sum

$$\sum_{m=0}^{\lfloor \gamma p \rfloor} \prod_{i=1}^{n} \frac{\alpha_i(\alpha_i+1)\cdots(\alpha_i+m-1)}{\beta_i(\beta_i+1)\cdots(\beta_i+m-1)} z^m \pmod{p^e}$$

for fixed $\alpha_i, \beta_i, \gamma \in \mathbb{Q}$. The sum has $O(p)$ terms.

But suppose we want this for all primes $p \leq X$. Then we can reinterpret the problem as computing, for $m = 0, 1, \ldots$, the sum up to a given m modulo the product of all $p \leq X$ for which $m \leq \lfloor \gamma p \rfloor$. There is a way to do this optimally using **remainder trees**[2] (Costa–Gerbicz–Harvey, Harvey–Sutherland); this is implemented generically in C by Sutherland.

[2] More precisely **remainder forests**, but in theory the difference only affects log factors. In practice it also greatly reduces memory usage.

A model: truncated hypergeometric sums

Computing the coefficient of p^{-s} in a hypergeometric L-function reduces (very roughly) to computing a truncated hypergeometric sum

$$\sum_{m=0}^{\lfloor \gamma p \rfloor} \prod_{i=1}^{n} \frac{\alpha_i(\alpha_i+1)\cdots(\alpha_i+m-1)}{\beta_i(\beta_i+1)\cdots(\beta_i+m-1)} z^m \pmod{p^e}$$

for fixed $\alpha_i, \beta_i, \gamma \in \mathbb{Q}$. The sum has $O(p)$ terms.

But suppose we want this for all primes $p \leq X$. Then we can reinterpret the problem as computing, for $m = 0, 1, \ldots$, the sum up to a given m modulo the product of all $p \leq X$ for which $m \leq \lfloor \gamma p \rfloor$. There is a way to do this optimally using **remainder trees**[2] (Costa–Gerbicz–Harvey, Harvey–Sutherland); this is implemented generically in C by Sutherland.

[2] More precisely **remainder forests**, but in theory the difference only affects log factors. In practice it also greatly reduces memory usage.

Accumulating remainder trees

Say we are given integers (or matrices) $A_0, \dots, A_{b-1}$ and integers $m_0, \dots, m_{b-1}$, 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 $m_0, \dots, m_{b-1}$, 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 $m_0, \dots, m_{b-1}$, and we want to compute simultaneously

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

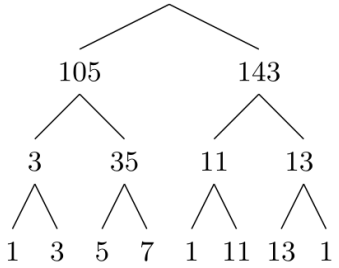
To simplify, assume $b = 2^\ell$. Form a complete binary tree of depth ℓ with nodes (i, j) where $i = 0, \dots, \ell$ and $j = 0, \dots, 2^{i-1}$. By computing from the leaves to the root, we can compute products over dyadic ranges:

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

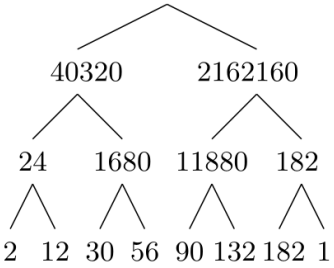
Then from the root to the leaves, we compute the products $C_{i,j} := A_{i,0} \cdots A_{i,j-1} \pmod{m_{i,j}}$ by writing

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

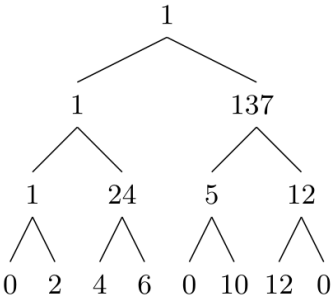
Illustration (Harvey–Sutherland, 2014)



$m_{i,j}$



$A_{i,j}$



$C_{i,j}$

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 $p \leq X$.

Back to truncated hypergeometric sums

By taking $A_i = \begin{pmatrix} g(i) & 0 \\ 1 & f(i+1)z \end{pmatrix}$, for any $\gamma \in \mathbb{Q} \cap (0,1]$ and e, we can efficiently compute for all $p \leq X$ the sums

$$\sum_{i=1}^{\lceil \gamma p \rceil} \frac{f(1)\cdots f(i-1)}{g(1)\cdots g(i-1)} z^{i-1} \pmod{p^e}.$$

By the way, the previous slide is a special case of this with $f(i) = i^j, g(i) = (i+1)^j$.

Fine print: the actual hypergeometric trace formula includes some p-adic analytic expressions (like evaluations of Γ_p and multiplicative[3] lifts) that don't fit this paradigm. One can handle these uniformly by (a variant) of an idea of Harvey: replace each matrix entry by a suitable block matrix, then perform a postcomputation for each p separately.

[3] a/k/a "Teichmüller", but I never liked this eponym

Back to truncated hypergeometric sums

By taking $A_i = \begin{pmatrix} g(i) & 0 \\ 1 & f(i+1)z \end{pmatrix}$, for any $\gamma \in \mathbb{Q} \cap (0,1]$ and e, we can efficiently compute for all $p \leq X$ the sums

$$\sum_{i=1}^{\lceil \gamma p \rceil} \frac{f(1)\cdots f(i-1)}{g(1)\cdots g(i-1)} z^{i-1} \pmod{p^e}.$$

By the way, the previous slide is a special case of this with $f(i) = i^j, g(i) = (i+1)^j$.

Fine print: the actual hypergeometric trace formula includes some p-adic analytic expressions (like evaluations of Γ_p and multiplicative[3] lifts) that don't fit this paradigm. One can handle these uniformly by (a variant) of an idea of Harvey: replace each matrix entry by a suitable block matrix, then perform a postcomputation for each p separately.

[3] a/k/a "Teichmüller", but I never liked this eponym

Back to truncated hypergeometric sums

By taking $A_i = \begin{pmatrix} g(i) & 0 \\ 1 & f(i+1)z \end{pmatrix}$, for any $\gamma \in \mathbb{Q} \cap (0,1]$ and e, we can efficiently compute for all $p \leq X$ the sums

$$\sum_{i=1}^{\lceil \gamma p \rceil} \frac{f(1)\cdots f(i-1)}{g(1)\cdots g(i-1)} z^{i-1} \pmod{p^e}.$$

By the way, the previous slide is a special case of this with $f(i) = i^j, g(i) = (i+1)^j$.

Fine print: the actual hypergeometric trace formula includes some p-adic analytic expressions (like evaluations of Γ_p and multiplicative[3] lifts) that don't fit this paradigm. One can handle these uniformly by (a variant) of an idea of Harvey: replace each matrix entry by a suitable block matrix, then perform a postcomputation for each p separately.

[3] a/k/a "Teichmüller", but I never liked this eponym

That QR code again