Computing Zeta Functions of Calabi-Yau Threefolds

MITP Physics and Number Theory, January 2025

Michael Stepniczka

Cornell University, Department of Physics

Joint work in progress with Pyry Kuusela and Mikey Lathwood

Calabi-Yau Manifolds

From Physics

- Let spacetime have a product structure $M_{10} = M_4 \times X_6$ with X_6 a compact 6-dimensional manifold
  - To solve the 10D-vacuum Einstein equations, we require that X_6 be Ricci flat
  - The only known examples are Calabi-Yau threefolds (CY3s)! In addition to being Ricci flat, the manifold is also Kähler
- Type IIB string theory contains data about even-dimensional forms. We can then study the structure using **complex algebraic geometry** (i.e. over $\mathbb{C}$)
  - More specifically, IIB comes with data including: 2-forms B_2, C_2 and corresponding field strengths $H_3 = dB_2, F_3 = dC_2$. Together with the axiodilaton τ, these are packaged as $G_3 = F_3 - \tau H_3$

Calabi-Yau Manifolds

From Physics

- Let spacetime have a product structure $M_{10} = M_4 \times X_6$ with X_6 a compact 6-dimensional manifold
  - To solve the 10D-vacuum Einstein equations, we require that X_6 be Ricci flat
  - The only known examples are Calabi-Yau threefolds (CY3s)! In addition to being Ricci flat, the manifold is also Kähler
- Type IIB string theory contains data about even-dimensional forms. We can then study the structure using **complex algebraic geometry** (i.e. over $\mathbb{C}$)
  - More specifically, IIB comes with data including: 2-forms B_2, C_2 and corresponding field strengths $H_3 = dB_2, F_3 = dC_2$. Together with the axiodilaton τ, these are packaged as $G_3 = F_3 - \tau H_3$

Calabi-Yau Manifolds

From Physics

- Let spacetime have a product structure $M_{10} = M_4 \times X_6$ with X_6 a compact 6-dimensional manifold
  - To solve the 10D-vacuum Einstein equations, we require that X_6 be Ricci flat
  - The only known examples are Calabi-Yau threefolds (CY3s)! In addition to being Ricci flat, the manifold is also Kähler
- Type IIB string theory contains data about even-dimensional forms. We can then study the structure using **complex algebraic geometry** (i.e. over $\mathbb{C}$)
  - More specifically, IIB comes with data including: 2-forms B_2, C_2 and corresponding field strengths $H_3 = dB_2, F_3 = dC_2$. Together with the axiodilaton τ, these are packaged as $G_3 = F_3 - \tau H_3$

Calabi-Yau Manifolds

From Physics

- Let spacetime have a product structure $M_{10} = M_4 \times X_6$ with X_6 a compact 6-dimensional manifold
  - To solve the 10D-vacuum Einstein equations, we require that X_6 be Ricci flat
  - The only known examples are Calabi-Yau threefolds (CY3s)! In addition to being Ricci flat, the manifold is also Kähler
- Type IIB string theory contains data about even-dimensional forms. We can then study the structure using **complex algebraic geometry** (i.e. over $\mathbb{C}$)
  - More specifically, IIB comes with data including: 2-forms B_2, C_2 and corresponding field strengths $H_3 = dB_2, F_3 = dC_2$. Together with the axiodilaton τ, these are packaged as $G_3 = F_3 - \tau H_3$

- Let spacetime have a product structure $M_{10} = M_4 \times X_6$ with X_6 a compact 6-dimensional manifold
  - To solve the 10D-vacuum Einstein equations, we require that X_6 be Ricci flat
  - The only known examples are Calabi-Yau threefolds (CY3s)! In addition to being Ricci flat, the manifold is also Kähler
- Type IIB string theory contains data about even-dimensional forms. We can then study the structure using **complex algebraic geometry** (i.e. over $\mathbb{C}$)
  - More specifically, IIB comes with data including: 2-forms B_2, C_2 and corresponding field strengths $H_3 = dB_2, F_3 = dC_2$. Together with the axiodilaton τ, these are packaged as $G_3 = F_3 - \tau H_3$

Calabi-Yau Manifolds

From Physics

- Better than complex algebraic geometry, the largest-known collection of CY3s comes from **toric geometry** [Batyrev; alg-geom/9310003]
  - CY3s are found as hypersurfaces in toric varieties
  - These toric varieties are obtained from (suitable triangulations of) 4D reflexive polytopes
    - **Combinatorial** in nature!
    - Question: How much of the CY data can be obtained purely from this combinatorial data?
  - Polytopal construction lends itself to computer-based studies via e.g. `CYTools`

- Better than complex algebraic geometry, the largest-known collection of CY3s comes from **toric geometry** [Batyrev; alg-geom/9310003]
  - CY3s are found as hypersurfaces in toric varieties
  - These toric varieties are obtained from (suitable triangulations of) 4D reflexive polytopes
    - **Combinatorial** in nature!
    - Question: How much of the CY data can be obtained purely from this combinatorial data?
  - Polytopal construction lends itself to computer-based studies via e.g. `CYTools`

Calabi-Yau Manifolds

From Physics

- Better than complex algebraic geometry, the largest-known collection of CY3s comes from **toric geometry** [Batyrev; alg-geom/9310003]
  - CY3s are found as hypersurfaces in toric varieties
  - These toric varieties are obtained from (suitable triangulations of) 4D reflexive polytopes
    - **Combinatorial** in nature!
    - Question: How much of the CY data can be obtained purely from this combinatorial data?
  - Polytopal construction lends itself to computer-based studies via e.g. `CYTools`

Calabi-Yau Manifolds

From Physics

- Better than complex algebraic geometry, the largest-known collection of CY3s comes from **toric geometry** [Batyrev; alg-geom/9310003]
  - CY3s are found as hypersurfaces in toric varieties
  - These toric varieties are obtained from (suitable triangulations of) 4D reflexive polytopes
    - **Combinatorial** in nature!
    - Question: How much of the CY data can be obtained purely from this combinatorial data?
  - Polytopal construction lends itself to computer-based studies via e.g. `CYTools`

Calabi-Yau Manifolds

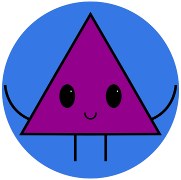
From Physics

- Better than complex algebraic geometry, the largest-known collection of CY3s comes from **toric geometry** [Batyrev; alg-geom/9310003]
  - CY3s are found as hypersurfaces in toric varieties
  - These toric varieties are obtained from (suitable triangulations of) 4D reflexive polytopes
    - **Combinatorial** in nature!
    - Question: How much of the CY data can be obtained purely from this combinatorial data?
  - Polytopal construction lends itself to computer-based studies via e.g. `CYTools`

Calabi-Yau Manifolds

From Physics

- Better than complex algebraic geometry, the largest-known collection of CY3s comes from **toric geometry** [Batyrev; alg-geom/9310003]
  - CY3s are found as hypersurfaces in toric varieties
  - These toric varieties are obtained from (suitable triangulations of) 4D reflexive polytopes
    - ⋆ **Combinatorial** in nature!
    - ⋆ Question: How much of the CY data can be obtained purely from this combinatorial data?
  - Polytopal construction lends itself to computer-based studies via e.g. `CYTools`



- More generally, a Calabi-Yau n-fold can be defined to be a compact Kähler manifold with vanishing first Chern class
- Examples:
  - $n = 1$: Complex Tori $\leftrightarrow$ Elliptic Curves
  - $n = 2$: K3 surfaces
  - $n = 3$: CY3s!
- As with elliptic curves, CYs can be studied via arithmetic geometry
  - Example: finding point counts over finite fields

Calabi-Yau Manifolds

From Mathematics

- More generally, a Calabi-Yau n-fold can be defined to be a compact Kähler manifold with vanishing first Chern class
- Examples:
  - $n = 1$: Complex Tori $\leftrightarrow$ **Elliptic Curves**
  - $n = 2$: K3 surfaces
  - $n = 3$: CY3s!
- As with elliptic curves, CYs can be studied via arithmetic geometry
  - Example: finding point counts over finite fields

Calabi-Yau Manifolds

From Mathematics

- More generally, a Calabi-Yau n-fold can be defined to be a compact Kähler manifold with vanishing first Chern class
- Examples:
  - $n = 1$: Complex Tori $\leftrightarrow$ **Elliptic Curves**
  - $n = 2$: K3 surfaces
  - $n = 3$: CY3s!
- As with elliptic curves, CYs can be studied via arithmetic geometry
  - Example: finding point counts over finite fields

Calabi-Yau Manifolds

From Mathematics

- More generally, a Calabi-Yau n-fold can be defined to be a compact Kähler manifold with vanishing first Chern class
- Examples:
  - $n = 1$: Complex Tori $\leftrightarrow$ **Elliptic Curves**
  - $n = 2$: K3 surfaces
  - $n = 3$: CY3s!
- As with elliptic curves, CYs can be studied via arithmetic geometry
  - Example: finding point counts over finite fields

- More generally, a Calabi-Yau n-fold can be defined to be a compact Kähler manifold with vanishing first Chern class
- Examples:
  - $n = 1$: Complex Tori $\leftrightarrow$ **Elliptic Curves**
  - $n = 2$: K3 surfaces
  - $n = 3$: CY3s!
- As with elliptic curves, CYs can be studied via arithmetic geometry
  - Example: finding point counts over finite fields

- More generally, a Calabi-Yau n-fold can be defined to be a compact Kähler manifold with vanishing first Chern class
- Examples:
  - $n = 1$: Complex Tori $\leftrightarrow$ **Elliptic Curves**
  - $n = 2$: K3 surfaces
  - $n = 3$: CY3s!
- As with elliptic curves, CYs can be studied via arithmetic geometry
  - Example: finding point counts over finite fields

- More generally, a Calabi-Yau n-fold can be defined to be a compact Kähler manifold with vanishing first Chern class
- Examples:
  - $n = 1$: Complex Tori $\leftrightarrow$ **Elliptic Curves**
  - $n = 2$: K3 surfaces
  - $n = 3$: CY3s!
- As with elliptic curves, CYs can be studied via arithmetic geometry
  - Example: finding point counts over finite fields

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
  - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
  - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
  - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
    - $a_3 = 3 + 1 - 4 = 0$
  - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
    - $a_5 = 5 + 1 - 4 = 2$
  - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
  - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
  - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
  - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
    - $a_3 = 3 + 1 - 4 = 0$
  - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
    - $a_5 = 5 + 1 - 4 = 2$
  - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
  - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
  - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
  - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
    - $a_3 = 3 + 1 - 4 = 0$
  - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
    - $a_5 = 5 + 1 - 4 = 2$
  - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
  - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
  - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
  - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
    - $a_3 = 3 + 1 - 4 = 0$
  - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
    - $a_5 = 5 + 1 - 4 = 2$
  - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
  - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
  - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
  - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
    - $a_3 = 3 + 1 - 4 = 0$
  - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
    - $a_5 = 5 + 1 - 4 = 2$
  - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
  - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
  - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
  - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
    - $a_3 = 3 + 1 - 4 = 0$
  - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
    - $a_5 = 5 + 1 - 4 = 2$
  - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
  - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
  - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
  - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
    - ⋆ $a_3 = 3 + 1 - 4 = 0$
  - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
    - ⋆ $a_5 = 5 + 1 - 4 = 2$
  - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Elliptic Curve Modularity

- Elliptic curves over $\mathbb{Q}$ can be written in Weierstrass form $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$
    - These curves can then be considered over finite field, e.g. $\mathbb{F}_p$
    - The number of solutions of the elliptic curve E over these finite fields are related to Fourier coefficients of a weight-2 modular form via $a_p = p + 1 - \#E(\mathbb{F}_p)$
- Example: $y^2 = x^3 + x$
    - Over $\mathbb{F}_3$, we have solutions $(x, y) = (0, 0), (2, 1), (2, 2)$, and a point at infinity
        - ⋆ $a_3 = 3 + 1 - 4 = 0$
    - Over $\mathbb{F}_5$, we have solutions $(x, y) = (0, 0), (2, 0), (3, 0)$, and a point at infinity
        - ⋆ $a_5 = 5 + 1 - 4 = 2$
    - Compare to **64.2.a.a**: $f(q) = q + 2q^5 - 3q^9 + \cdots$
- Modularity then says elliptic curves $E/\mathbb{Q}$ have associated to them such modular forms f.

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
  - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
  - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
  - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
  - Singular K3s
  - Rigid CY3s
  - SUSY Flux Vacua

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
  - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
  - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
  - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
  - Singular K3s
  - Rigid CY3s
  - SUSY Flux Vacua

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
  - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
  - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
  - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
  - Singular K3s
  - Rigid CY3s
  - SUSY Flux Vacua

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
  - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
  - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
  - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
  - Singular K3s
  - Rigid CY3s
  - SUSY Flux Vacua

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
  - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
  - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
  - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
  - Singular K3s
  - Rigid CY3s
  - SUSY Flux Vacua

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
  - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
  - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
  - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
  - Singular K3s
  - Rigid CY3s
  - SUSY Flux Vacua

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
  - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
  - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
  - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
  - Singular K3s
  - Rigid CY3s
  - SUSY Flux Vacua

Towards Higher Modularity

- More generally, this story can be told in terms of representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$:
    - The middle cohomology of an elliptic curve $E/\mathbb{Q}$ provides us with a 2-dimensional representation $\rho_2(E)$.
    - On the other hand, cusp forms f that are simultaneous eigenvectors of all Hecke operators are also associated to 2-dimensional representations $\rho_2(f)$
    - It is then the modularity theorem which associates to every $E/\mathbb{Q}$ such an f for which the representations coincide
- This story can be generalized to Calabi-Yau n-folds X. Modularity can arise, for instance, when the representation decomposes into a 2-dimensional subrepresentation and a $(b_n - 2)$-dimensional component.
    - Singular K3s
    - Rigid CY3s
    - SUSY Flux Vacua

Let $X/\mathbb{Q}$ be a smooth projective variety. As before, we can clear denominators and view $X/\mathbb{Z}$. Furthermore, we can reduce modulo p to view this as $X/\mathbb{F}_p$. As $\mathbb{F}_p$ is a subfield of $\mathbb{F}_{p^n}$, we can finally study $X/\mathbb{F}_{p^n}$.

Definition

For $X/\mathbb{Q}$ as above and for good primes p, define the *local zeta function* to be:

$$\zeta_p(X,T) = \exp\left(\sum_{n=1}^{\infty} \frac{\#X(\mathbb{F}_{p^n})T^n}{n}\right),$$

with $\#X(\mathbb{F}_{p^n})$ denoting the point count over $\mathbb{F}_{p^n}$.

Let $X/\mathbb{Q}$ be a smooth projective variety. As before, we can clear denominators and view $X/\mathbb{Z}$. Furthermore, we can reduce modulo p to view this as $X/\mathbb{F}_p$. As $\mathbb{F}_p$ is a subfield of $\mathbb{F}_{p^n}$, we can finally study $X/\mathbb{F}_{p^n}$.

Definition

For $X/\mathbb{Q}$ as above and for good primes p, define the *local zeta function* to be:

$$\zeta_p(X,T) = \exp\left(\sum_{n=1}^{\infty} \frac{\#X(\mathbb{F}_{p^n})T^n}{n}\right),$$

with $\#X(\mathbb{F}_{p^n})$ denoting the point count over $\mathbb{F}_{p^n}$.

Example

Let X be a point. Clearing denominators, $\#X(\mathbb{F}_{p^n}) = 1$. Then:

$$\zeta_p(\{\mathrm{pt}\}, T) = \exp\left(\sum_{n=1}^{\infty} \frac{T^n}{n}\right) = \frac{1}{1-T}$$

From the local zeta functions, construct:

$$\zeta(X, s) = \prod_p \zeta_p(X, p^{-s}),$$

and find the usual Riemann zeta function

$$\zeta(\{\mathrm{pt}\}, s) = \prod_p \frac{1}{1-p^{-s}} = \zeta(s).$$

Theorem (Rationality)

$\zeta_p(X,T)$ is a rational function of T. Specifically:

$$\zeta_p(X,T) = \frac{R_p^{(1)}(X,T)\cdots R_p^{(2n-1)}(X,T)}{R_p^{(0)}(X,T)\cdots R_p^{(2n)}(X,T)},$$

where each $R_p^{(i)}(T)$ has integral coefficients and is of degree b_i, the i'th Betti number.

There are other nice properties that these zeta functions satisfy: namely, a functional equation and a Riemann hypothesis.

To prove this, Dwork used p-adic cohomological techniques.

- First, the numbers $\#X(\mathbb{F}_p)$ can be determined as follows:
  - Recall the Frobenius endomorphism:

$$\mathrm{Frob}_p : (x_1, \ldots, x_k) \mapsto (x_1^p, \ldots, x_k^p)$$

  - As $x^p \equiv x \bmod p$ for all $x \in \mathbb{F}_p$, the fixed points of Frob_p over any field extension are exactly those in $\mathbb{F}_p$
  - You can repeat this for $\mathbb{F}_{p^n}$
- Now think of a family of CY3s X_φ. One can define a p-adic cohomology theory $H^k(X_\varphi, \mathbb{Q}_p)$ and pullback the Frobenius map:

$$\mathrm{Fr}_p : H^k(X_\varphi, \mathbb{Q}_p) \to H^k(X_\varphi, \mathbb{Q}_p)$$

- Apply the Lefschetz fixed point theorem:

$$\#X_\varphi(\mathbb{F}_{p^n}) = \sum_{m=0}^{6} (-1)^m \mathrm{Tr}(\mathrm{Fr}_{p^n}|_{H^m(X_\varphi, \mathbb{Q}_p)})$$

To prove this, Dwork used p-adic cohomological techniques.

- First, the numbers $\#X(\mathbb{F}_p)$ can be determined as follows:
  - Recall the Frobenius endomorphism:
    $$\mathrm{Frob}_p : (x_1, \dots, x_k) \mapsto (x_1^p, \dots, x_k^p)$$
  - As $x^p \equiv x \bmod p$ for all $x \in \mathbb{F}_p$, the fixed points of Frob_p over any field extension are exactly those in $\mathbb{F}_p$
  - You can repeat this for $\mathbb{F}_{p^n}$
- Now think of a family of CY3s X_φ. One can define a p-adic cohomology theory $H^k(X_\varphi, \mathbb{Q}_p)$ and pullback the Frobenius map:
  $$\mathrm{Fr}_p : H^k(X_\varphi, \mathbb{Q}_p) \to H^k(X_\varphi, \mathbb{Q}_p)$$
- Apply the Lefschetz fixed point theorem:
  $$\#X_\varphi(\mathbb{F}_{p^n}) = \sum_{m=0}^{6} (-1)^m \mathrm{Tr}(\mathrm{Fr}_{p^n}|_{H^m(X_\varphi, \mathbb{Q}_p)})$$

To prove this, Dwork used p-adic cohomological techniques.

- First, the numbers $\#X(\mathbb{F}_p)$ can be determined as follows:
  - Recall the Frobenius endomorphism:

$$\mathrm{Frob}_p : (x_1, \ldots, x_k) \mapsto (x_1^p, \ldots, x_k^p)$$

  - As $x^p \equiv x \bmod p$ for all $x \in \mathbb{F}_p$, the fixed points of Frob_p over any field extension are exactly those in $\mathbb{F}_p$
  - You can repeat this for $\mathbb{F}_{p^n}$
- Now think of a family of CY3s X_φ. One can define a p-adic cohomology theory $H^k(X_\varphi, \mathbb{Q}_p)$ and pullback the Frobenius map:

$$\mathrm{Fr}_p : H^k(X_\varphi, \mathbb{Q}_p) \to H^k(X_\varphi, \mathbb{Q}_p)$$

- Apply the Lefschetz fixed point theorem:

$$\#X_\varphi(\mathbb{F}_{p^n}) = \sum_{m=0}^{6} (-1)^m \mathrm{Tr}(\mathrm{Fr}_{p^n}|_{H^m(X_\varphi, \mathbb{Q}_p)})$$

To prove this, Dwork used p-adic cohomological techniques.

- First, the numbers $\#X(\mathbb{F}_p)$ can be determined as follows:
  - Recall the Frobenius endomorphism:

$$\mathrm{Frob}_p : (x_1, \ldots, x_k) \mapsto (x_1^p, \ldots, x_k^p)$$

  - As $x^p \equiv x \bmod p$ for all $x \in \mathbb{F}_p$, the fixed points of Frob_p over any field extension are exactly those in $\mathbb{F}_p$
  - You can repeat this for $\mathbb{F}_{p^n}$
- Now think of a family of CY3s X_φ. One can define a p-adic cohomology theory $H^k(X_\varphi, \mathbb{Q}_p)$ and pullback the Frobenius map:

$$\mathrm{Fr}_p : H^k(X_\varphi, \mathbb{Q}_p) \to H^k(X_\varphi, \mathbb{Q}_p)$$

- Apply the Lefschetz fixed point theorem:

$$\#X_\varphi(\mathbb{F}_{p^n}) = \sum_{m=0}^{6} (-1)^m \mathrm{Tr}(\mathrm{Fr}_{p^n}|_{H^m(X_\varphi, \mathbb{Q}_p)})$$

To prove this, Dwork used p-adic cohomological techniques.

- First, the numbers $\#X(\mathbb{F}_p)$ can be determined as follows:
  - Recall the Frobenius endomorphism:

$$\mathrm{Frob}_p : (x_1, \ldots, x_k) \mapsto (x_1^p, \ldots, x_k^p)$$

  - As $x^p \equiv x \bmod p$ for all $x \in \mathbb{F}_p$, the fixed points of Frob_p over any field extension are exactly those in $\mathbb{F}_p$
  - You can repeat this for $\mathbb{F}_{p^n}$
- Now think of a family of CY3s X_φ. One can define a p-adic cohomology theory $H^k(X_\varphi, \mathbb{Q}_p)$ and pullback the Frobenius map:

$$\mathrm{Fr}_p : H^k(X_\varphi, \mathbb{Q}_p) \to H^k(X_\varphi, \mathbb{Q}_p)$$

- Apply the Lefschetz fixed point theorem:

$$\#X_\varphi(\mathbb{F}_{p^n}) = \sum_{m=0}^{6} (-1)^m \mathrm{Tr}(\mathrm{Fr}_{p^n}|_{H^m(X_\varphi, \mathbb{Q}_p)})$$

To prove this, Dwork used p-adic cohomological techniques.

- First, the numbers $\#X(\mathbb{F}_p)$ can be determined as follows:
  - Recall the Frobenius endomorphism:

$$\mathrm{Frob}_p : (x_1, \ldots, x_k) \mapsto (x_1^p, \ldots, x_k^p)$$

  - As $x^p \equiv x \bmod p$ for all $x \in \mathbb{F}_p$, the fixed points of Frob_p over any field extension are exactly those in $\mathbb{F}_p$
  - You can repeat this for $\mathbb{F}_{p^n}$
- Now think of a family of CY3s X_φ. One can define a p-adic cohomology theory $H^k(X_\varphi, \mathbb{Q}_p)$ and pullback the Frobenius map:

$$\mathrm{Fr}_p : H^k(X_\varphi, \mathbb{Q}_p) \to H^k(X_\varphi, \mathbb{Q}_p)$$

- Apply the Lefschetz fixed point theorem:

$$\#X_\varphi(\mathbb{F}_{p^n}) = \sum_{m=0}^{6} (-1)^m \mathrm{Tr}(\mathrm{Fr}_{p^n}|_{H^m(X_\varphi, \mathbb{Q}_p)})$$

To prove this, Dwork used p-adic cohomological techniques.

- First, the numbers $\#X(\mathbb{F}_p)$ can be determined as follows:
  - Recall the Frobenius endomorphism:

$$\mathrm{Frob}_p : (x_1, \ldots, x_k) \mapsto (x_1^p, \ldots, x_k^p)$$

  - As $x^p \equiv x \mod p$ for all $x \in \mathbb{F}_p$, the fixed points of Frob_p over any field extension are exactly those in $\mathbb{F}_p$
  - You can repeat this for $\mathbb{F}_{p^n}$
- Now think of a family of CY3s X_φ. One can define a p-adic cohomology theory $H^k(X_\varphi, \mathbb{Q}_p)$ and pullback the Frobenius map:

$$\mathrm{Fr}_p : H^k(X_\varphi, \mathbb{Q}_p) \to H^k(X_\varphi, \mathbb{Q}_p)$$

- Apply the Lefschetz fixed point theorem:

$$\#X_\varphi(\mathbb{F}_{p^n}) = \sum_{m=0}^{6} (-1)^m \mathrm{Tr}(\mathrm{Fr}_{p^n}|_{H^m(X_\varphi, \mathbb{Q}_p)})$$

- The characteristic polynomial of the inverse of the Frobenius map is then:

$$\begin{aligned} R_p^{(m)}(X_\varphi, T) &= \det(I - T\mathrm{Fr}_p^{-1}|_{H^m(X_\varphi,\mathbb{Q}_p)}) \\ &= \det(I - TU_p(\varphi)) \end{aligned}$$

- The Hodge diamond for a CY3s is:

$$\begin{array}{ccccccc} & & & 1 & & & \\ & & 0 & & 0 & & \\ & 0 & & h^{1,1} & & 0 & \\ 1 & & h^{2,1} & & h^{2,1} & & 1 \\ & 0 & & h^{1,1} & & 0 & \\ & & 0 & & 0 & & \\ & & & 1 & & & \end{array}$$

- This tells us $b_0 = b_6 = 1$, $b_1 = b_5 = 0$, $b_2 = b_4 = h^{1,1}$, and $b_3 = 2 + 2h^{2,1}$.

- The characteristic polynomial of the inverse of the Frobenius map is then:

$$\begin{aligned} R_p^{(m)}(X_\varphi, T) &= \det(I - T\mathrm{Fr}_p^{-1}|_{H^m(X_\varphi,\mathbb{Q}_p)}) \\ &= \det(I - TU_p(\varphi)) \end{aligned}$$

- The Hodge diamond for a CY3s is:

$$\begin{array}{ccccccc} & & & 1 & & & \\ & & 0 & & 0 & & \\ & 0 & & h^{1,1} & & 0 & \\ 1 & & h^{2,1} & & h^{2,1} & & 1 \\ & 0 & & h^{1,1} & & 0 & \\ & & 0 & & 0 & & \\ & & & 1 & & & \end{array}$$

- This tells us $b_0 = b_6 = 1$, $b_1 = b_5 = 0$, $b_2 = b_4 = h^{1,1}$, and $b_3 = 2 + 2h^{2,1}$.

- The characteristic polynomial of the inverse of the Frobenius map is then:

$$\begin{aligned} R_p^{(m)}(X_\varphi, T) &= \det(I - T\mathrm{Fr}_p^{-1}|_{H^m(X_\varphi,\mathbb{Q}_p)}) \\ &= \det(I - TU_p(\varphi)) \end{aligned}$$

- The Hodge diamond for a CY3s is:

$$\begin{array}{ccccccc} & & & 1 & & & \\ & & 0 & & 0 & & \\ & 0 & & h^{1,1} & & 0 & \\ 1 & & h^{2,1} & & h^{2,1} & & 1 \\ & 0 & & h^{1,1} & & 0 & \\ & & 0 & & 0 & & \\ & & & 1 & & & \end{array}$$

- This tells us $b_0 = b_6 = 1$, $b_1 = b_5 = 0$, $b_2 = b_4 = h^{1,1}$, and $b_3 = 2 + 2h^{2,1}$.

- Now our zeta function is:

$$\begin{aligned}\zeta_p(X_\varphi, T) &= \frac{R_p^{(1)}(X_\varphi, T) R_p^{(3)}(X_\varphi, T) R_p^{(5)}(X_\varphi, T)}{R_p^{(0)}(X_\varphi, T) R_p^{(2)}(X_\varphi, T) R_p^{(4)}(X_\varphi, T) R_p^{(6)}(X_\varphi, T)} \\ &= \frac{R_p^{(3)}(X_\varphi, T)}{(1-T)(1-pT)^{h^{1,1}}(1-p^2T)^{h^{1,1}}(1-p^3T)},\end{aligned}$$

  where I have glossed over a few details.
- In this case, we only need to care about $R_p^{(3)}(X_\varphi, T)$
  - We'll get back to this point later!

- Now our zeta function is:

$$\begin{aligned}\zeta_p(X_\varphi, T) &= \frac{R_p^{(1)}(X_\varphi, T) R_p^{(3)}(X_\varphi, T) R_p^{(5)}(X_\varphi, T)}{R_p^{(0)}(X_\varphi, T) R_p^{(2)}(X_\varphi, T) R_p^{(4)}(X_\varphi, T) R_p^{(6)}(X_\varphi, T)} \\ &= \frac{R_p^{(3)}(X_\varphi, T)}{(1-T)(1-pT)^{h^{1,1}}(1-p^2T)^{h^{1,1}}(1-p^3T)},\end{aligned}$$

where I have glossed over a few details.

- In this case, we only need to care about $R_p^{(3)}(X_\varphi, T)$
    - We'll get back to this point later!

- BH modularity/rank-2 attractor points [e.g. Candelas, de la Ossa, Elmi, van Straten 1912.06146v1]
- Supersymmetric flux vacua and F-theory/M-theory modularity [e.g. Kachru, Nally, Yang 2001.06022v2, 2010.07285v2; Jockers, Kotlewski, Kuusela 2312.07611v3]
- Ratio of L-functions to GWs [Candelas, de la Ossa, McGovern 2410.07107v1]

$$\frac{3\pi}{2}\frac{L_{\mathbf{54.4.a.c}}(1)}{L_{\mathbf{54.4.a.c}}(2)} = \sqrt{69} - \sqrt{\frac{2}{\pi^3}} \sum_{\substack{j\in\mathbb{Z}_{>0} \\ \mathfrak{p}\in\mathrm{pt}(j)}} (-1)^j \widetilde{N}_{\mathfrak{p}} \left(\frac{j}{3\pi\sqrt{69}}\right)^{l(\mathfrak{p})-1/2} * K_{l(\mathfrak{p})-1/2}\left(\frac{\pi j\sqrt{69}}{3}\right)$$

- Arithmetic geometry finds points of interest in complex geometry?

- BH modularity/rank-2 attractor points [e.g. Candelas, de la Ossa, Elmi, van Straten 1912.06146v1]
- Supersymmetric flux vacua and F-theory/M-theory modularity [e.g. Kachru, Nally, Yang 2001.06022v2, 2010.07285v2; Jockers, Kotlewski, Kuusela 2312.07611v3]
- Ratio of L-functions to GWs [Candelas, de la Ossa, McGovern 2410.07107v1]

$$\frac{3\pi}{2}\frac{L_{\mathbf{54.4.a.c}}(1)}{L_{\mathbf{54.4.a.c}}(2)} = \sqrt{69} - \sqrt{\frac{2}{\pi^3}} \sum_{\substack{j\in\mathbb{Z}_{>0} \\ \mathfrak{p}\in\mathrm{pt}(j)}} (-1)^j \widetilde{N}_{\mathfrak{p}} \left(\frac{j}{3\pi\sqrt{69}}\right)^{l(\mathfrak{p})-1/2} * K_{l(\mathfrak{p})-1/2}\left(\frac{\pi j\sqrt{69}}{3}\right)$$

- Arithmetic geometry finds points of interest in complex geometry?

- BH modularity/rank-2 attractor points [e.g. Candelas, de la Ossa, Elmi, van Straten 1912.06146v1]
- Supersymmetric flux vacua and F-theory/M-theory modularity [e.g. Kachru, Nally, Yang 2001.06022v2, 2010.07285v2; Jockers, Kotlewski, Kuusela 2312.07611v3]
- Ratio of L-functions to GWs [Candelas, de la Ossa, McGovern 2410.07107v1]

$$\frac{3\pi}{2}\frac{L_{\mathbf{54.4.a.c}}(1)}{L_{\mathbf{54.4.a.c}}(2)} = \sqrt{69} - \sqrt{\frac{2}{\pi^3}} \sum_{\substack{j\in\mathbb{Z}_{>0} \\ \mathfrak{p}\in\mathrm{pt}(j)}} (-1)^j \widetilde{N}_{\mathfrak{p}} \left(\frac{j}{3\pi\sqrt{69}}\right)^{l(\mathfrak{p})-1/2} * K_{l(\mathfrak{p})-1/2}\left(\frac{\pi j\sqrt{69}}{3}\right)$$

- Arithmetic geometry finds points of interest in complex geometry?

- BH modularity/rank-2 attractor points [e.g. Candelas, de la Ossa, Elmi, van Straten 1912.06146v1]
- Supersymmetric flux vacua and F-theory/M-theory modularity [e.g. Kachru, Nally, Yang 2001.06022v2, 2010.07285v2; Jockers, Kotlewski, Kuusela 2312.07611v3]
- Ratio of L-functions to GWs [Candelas, de la Ossa, McGovern 2410.07107v1]

$$\frac{3\pi}{2}\frac{L_{\mathbf{54.4.a.c}}(1)}{L_{\mathbf{54.4.a.c}}(2)} = \sqrt{69} - \sqrt{\frac{2}{\pi^3}} \sum_{\substack{j\in\mathbb{Z}_{>0}\\ \mathfrak{p}\in\mathrm{pt}(j)}} (-1)^j \widetilde{N}_{\mathfrak{p}} \left(\frac{j}{3\pi\sqrt{69}}\right)^{l(\mathfrak{p})-1/2} * K_{l(\mathfrak{p})-1/2}\left(\frac{\pi j\sqrt{69}}{3}\right)$$

- Arithmetic geometry finds points of interest in complex geometry?

Computing Zeta Functions

Method: Point Counts

- Description: Work from the definition of the zeta function

$$\zeta_p(X_\varphi, T) = \exp\left(\sum_{n=1}^{\infty} \frac{\#X_\varphi(\mathbb{F}_{p^n})T^n}{n}\right)$$

  and compute these point counts explicitly.
- Advantages:
  - Straightforward to implement
- Disadvantages:
  - Slow
  - Fixes a point in CS moduli space

- Description: Work from the definition of the zeta function

$$\zeta_p(X_\varphi, T) = \exp\left(\sum_{n=1}^{\infty} \frac{\#X_\varphi(\mathbb{F}_{p^n})T^n}{n}\right)$$

  and compute these point counts explicitly.
- Advantages:
  - Straightforward to implement
- Disadvantages:
  - Slow
  - Fixes a point in CS moduli space

- Description: Work from the definition of the zeta function

$$\zeta_p(X_\varphi, T) = \exp\left(\sum_{n=1}^{\infty} \frac{\# X_\varphi(\mathbb{F}_{p^n}) T^n}{n}\right)$$

  and compute these point counts explicitly.
- Advantages:
  - Straightforward to implement
- Disadvantages:
  - Slow
  - Fixes a point in CS moduli space

- See e.g. 0705.2056v1; 2104.07816v1; 2405.08067v1. (Candelas, de la Ossa, Kuusela, van Straten)
- As mentioned earlier, we have reduced the computation of the zeta function to the computation of $R_p^{(3)}(X_\varphi, T) = \det(I - TU_p(\varphi))$, where $U_p(\varphi)$ is the action of Fr_p^{-1} on the middle cohomology $H^3(X_\varphi, \mathbb{Q}_p)$.
  - Therefore, if we can determine $U_p(\varphi)$, we can compute the zeta function!
- Due to Dwork, $U_p(\varphi) = E(\varphi^p)^{-1} U_p(0) E(\varphi)$, where E is a matrix depending on the periods
  - We like 0 – LCS point!
  - Subtleties exist: if we chose φ such that $\varphi^p = \varphi$, then it would seem that ζ is independent of φ
  - Instead, it turns out that the periods converge when $||\varphi||_p < 1$, and we need to follow p-adic analytic continuation to define $U(\varphi)$ for $||\varphi||_p = 1$

- See e.g. 0705.2056v1; 2104.07816v1; 2405.08067v1. (Candelas, de la Ossa, Kuusela, van Straten)
- As mentioned earlier, we have reduced the computation of the zeta function to the computation of $R_p^{(3)}(X_\varphi, T) = \det(I - TU_p(\varphi))$, where $U_p(\varphi)$ is the action of Fr_p^{-1} on the middle cohomology $H^3(X_\varphi, \mathbb{Q}_p)$.
    - Therefore, if we can determine $U_p(\varphi)$, we can compute the zeta function!
- Due to Dwork, $U_p(\varphi) = E(\varphi^p)^{-1} U_p(0) E(\varphi)$, where E is a matrix depending on the periods
    - We like 0 – LCS point!
    - Subtleties exist: if we chose φ such that $\varphi^p = \varphi$, then it would seem that ζ is independent of φ
    - Instead, it turns out that the periods converge when $||\varphi||_p < 1$, and we need to follow p-adic analytic continuation to define $U(\varphi)$ for $||\varphi||_p = 1$

- See e.g. 0705.2056v1; 2104.07816v1; 2405.08067v1. (Candelas, de la Ossa, Kuusela, van Straten)
- As mentioned earlier, we have reduced the computation of the zeta function to the computation of $R_p^{(3)}(X_\varphi, T) = \det(I - TU_p(\varphi))$, where $U_p(\varphi)$ is the action of Fr_p^{-1} on the middle cohomology $H^3(X_\varphi, \mathbb{Q}_p)$.
  - Therefore, if we can determine $U_p(\varphi)$, we can compute the zeta function!
- Due to Dwork, $U_p(\varphi) = E(\varphi^p)^{-1}U_p(0)E(\varphi)$, where E is a matrix depending on the periods
  - We like 0 – LCS point!
  - Subtleties exist: if we chose φ such that $\varphi^p = \varphi$, then it would seem that ζ is independent of φ
  - Instead, it turns out that the periods converge when $||\varphi||_p < 1$, and we need to follow p-adic analytic continuation to define $U(\varphi)$ for $||\varphi||_p = 1$

- See e.g. 0705.2056v1; 2104.07816v1; 2405.08067v1. (Candelas, de la Ossa, Kuusela, van Straten)
- As mentioned earlier, we have reduced the computation of the zeta function to the computation of $R_p^{(3)}(X_\varphi, T) = \det(I - TU_p(\varphi))$, where $U_p(\varphi)$ is the action of Fr_p^{-1} on the middle cohomology $H^3(X_\varphi, \mathbb{Q}_p)$.
    - Therefore, if we can determine $U_p(\varphi)$, we can compute the zeta function!
- Due to Dwork, $U_p(\varphi) = E(\varphi^p)^{-1}U_p(0)E(\varphi)$, where E is a matrix depending on the periods
    - We like 0 – LCS point!
    - Subtleties exist: if we chose φ such that $\varphi^p = \varphi$, then it would seem that ζ is independent of φ
    - Instead, it turns out that the periods converge when $||\varphi||_p < 1$, and we need to follow p-adic analytic continuation to define $U(\varphi)$ for $||\varphi||_p = 1$

- See e.g. 0705.2056v1; 2104.07816v1; 2405.08067v1. (Candelas, de la Ossa, Kuusela, van Straten)
- As mentioned earlier, we have reduced the computation of the zeta function to the computation of $R_p^{(3)}(X_\varphi, T) = \det(I - TU_p(\varphi))$, where $U_p(\varphi)$ is the action of Fr_p^{-1} on the middle cohomology $H^3(X_\varphi, \mathbb{Q}_p)$.
  - Therefore, if we can determine $U_p(\varphi)$, we can compute the zeta function!
- Due to Dwork, $U_p(\varphi) = E(\varphi^p)^{-1}U_p(0)E(\varphi)$, where E is a matrix depending on the periods
  - We like 0 – LCS point!
  - Subtleties exist: if we chose φ such that $\varphi^p = \varphi$, then it would seem that ζ is independent of φ
  - Instead, it turns out that the periods converge when $||\varphi||_p < 1$, and we need to follow p-adic analytic continuation to define $U(\varphi)$ for $||\varphi||_p = 1$

Computing Zeta Functions

Method: Dwork Deformation

- See e.g. 0705.2056v1; 2104.07816v1; 2405.08067v1. (Candelas, de la Ossa, Kuusela, van Straten)
- As mentioned earlier, we have reduced the computation of the zeta function to the computation of $R_p^{(3)}(X_\varphi, T) = \det(I - TU_p(\varphi))$, where $U_p(\varphi)$ is the action of Fr_p^{-1} on the middle cohomology $H^3(X_\varphi, \mathbb{Q}_p)$.
    - Therefore, if we can determine $U_p(\varphi)$, we can compute the zeta function!
- Due to Dwork, $U_p(\varphi) = E(\varphi^p)^{-1} U_p(0) E(\varphi)$, where E is a matrix depending on the periods
    - We like 0 – LCS point!
    - Subtleties exist: if we chose φ such that $\varphi^p = \varphi$, then it would seem that ζ is independent of φ
    - Instead, it turns out that the periods converge when $||\varphi||_p < 1$, and we need to follow p-adic analytic continuation to define $U(\varphi)$ for $||\varphi||_p = 1$

- See e.g. 0705.2056v1; 2104.07816v1; 2405.08067v1. (Candelas, de la Ossa, Kuusela, van Straten)
- As mentioned earlier, we have reduced the computation of the zeta function to the computation of $R_p^{(3)}(X_\varphi, T) = \det(I - TU_p(\varphi))$, where $U_p(\varphi)$ is the action of Fr_p^{-1} on the middle cohomology $H^3(X_\varphi, \mathbb{Q}_p)$.
    - Therefore, if we can determine $U_p(\varphi)$, we can compute the zeta function!
- Due to Dwork, $U_p(\varphi) = E(\varphi^p)^{-1}U_p(0)E(\varphi)$, where E is a matrix depending on the periods
    - We like 0 – LCS point!
    - Subtleties exist: if we chose φ such that $\varphi^p = \varphi$, then it would seem that ζ is independent of φ
    - Instead, it turns out that the periods converge when $||\varphi||_p < 1$, and we need to follow p-adic analytic continuation to define $U(\varphi)$ for $||\varphi||_p = 1$

- Now the problem has been (largely) reduced to solving for the periods ϖ (\varpi) of our CY3.
- Recall that $\dim H^3 = b_3 = 2 + 2h^{2,1}$. Our holomorphic top form: $\Omega \in H^{(3,0)}(X_\varphi, \mathbb{C})$, and Griffiths transversality tells us:

$$
\begin{aligned}
\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\partial_{\varphi^j}\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}) \oplus H^{(1,2)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\partial_{\varphi^j}\partial_{\varphi^k}\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}) \oplus H^{(1,2)}(X_\varphi, \mathbb{C}) \oplus H^{(0,3)}(X_\varphi, \mathbb{C})
\end{aligned}
$$

- As the space is $2 + 2h^{2,1}$-dimensional, choosing a basis then gives us relations between the other derivatives and these basis elements. This then gives a system of differential equations that Ω satisfies, the *Picard-Fuchs* system.

- Now the problem has been (largely) reduced to solving for the periods ϖ (\varpi) of our CY3.
- Recall that $\dim H^3 = b_3 = 2 + 2h^{2,1}$. Our holomorphic top form: $\Omega \in H^{(3,0)}(X_\varphi, \mathbb{C})$, and Griffiths transversality tells us:

$$\begin{aligned}
\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\partial_{\varphi^j}\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}) \oplus H^{(1,2)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\partial_{\varphi^j}\partial_{\varphi^k}\Omega &\in H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}) \oplus H^{(1,2)}(X_\varphi, \mathbb{C}) \oplus H^{(0,3)}(X_\varphi, \mathbb{C})
\end{aligned}$$

- As the space is $2 + 2h^{2,1}$-dimensional, choosing a basis then gives us relations between the other derivatives and these basis elements. This then gives a system of differential equations that Ω satisfies, the *Picard-Fuchs* system.

- Now the problem has been (largely) reduced to solving for the periods ϖ (\varpi) of our CY3.
- Recall that $\dim H^3 = b_3 = 2 + 2h^{2,1}$. Our holomorphic top form: $\Omega \in H^{(3,0)}(X_\varphi, \mathbb{C})$, and Griffiths transversality tells us:

$$\begin{array}{rcl}
\Omega & \in & H^{(3,0)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\Omega & \in & H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\partial_{\varphi^j}\Omega & \in & H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}) \oplus H^{(1,2)}(X_\varphi, \mathbb{C}), \\
\partial_{\varphi^i}\partial_{\varphi^j}\partial_{\varphi^k}\Omega & \in & H^{(3,0)}(X_\varphi, \mathbb{C}) \oplus H^{(2,1)}(X_\varphi, \mathbb{C}) \oplus H^{(1,2)}(X_\varphi, \mathbb{C}) \oplus H^{(0,3)}(X_\varphi, \mathbb{C})
\end{array}$$

- As the space is $2 + 2h^{2,1}$-dimensional, choosing a basis then gives us relations between the other derivatives and these basis elements. This then gives a system of differential equations that Ω satisfies, the *Picard-Fuchs* system.

- Integrating, this can then be stated in terms of the *periods*; namely, $\mathcal{L}\varpi = 0$.
- These may be seen in different bases. One important one for our purposes is, extracting around the LCS point $\varphi = 0$, the Frobenius basis:

$$\varpi = \begin{pmatrix} f \\ f^i + fl^i \\ \frac{1}{2!}Y_{ijk}(f^{ij} + 2f^j l^k + f l^j l^k) \\ \frac{1}{3!}Y_{ijk}(f^{ijk} + 3f^{ij}l^k + 3f^i l^j l^k + f l^i l^j l^k) \end{pmatrix}$$

- Defining $(\vartheta_0, \vartheta_i, \vartheta^i, \vartheta^0) = \left(1, \theta_i, \hat{Y}^{ijk}\theta_j\theta_k, \frac{\hat{Y}^{ijk}}{m}\theta_i\theta_j\theta_k\right)$, we then have

$$E(\varphi)_a^b = \begin{pmatrix} \vartheta_a\varpi^b & \vartheta^a\varpi^b \\ \vartheta_a\varpi_b & \vartheta^a\varpi_b \end{pmatrix}$$

- Integrating, this can then be stated in terms of the *periods*; namely, $\mathcal{L}\varpi = 0$.
- These may be seen in different bases. One important one for our purposes is, extracting around the LCS point $\varphi = 0$, the Frobenius basis:

$$\varpi = \begin{pmatrix} f \\ f^i + fl^i \\ \frac{1}{2!}Y_{ijk}(f^{ij} + 2f^j l^k + fl^j l^k) \\ \frac{1}{3!}Y_{ijk}(f^{ijk} + 3f^{ij}l^k + 3f^i l^j l^k + fl^i l^j l^k) \end{pmatrix}$$

- Defining $(\vartheta_0, \vartheta_i, \vartheta^i, \vartheta^0) = \left(1, \theta_i, \hat{Y}^{ijk}\theta_j\theta_k, \frac{\hat{Y}^{ijk}}{m}\theta_i\theta_j\theta_k\right)$, we then have

$$E(\varphi)_a^b = \begin{pmatrix} \vartheta_a\varpi^b & \vartheta^a\varpi^b \\ \vartheta_a\varpi_b & \vartheta^a\varpi_b \end{pmatrix}$$

- Integrating, this can then be stated in terms of the *periods*; namely, $\mathcal{L}\varpi = 0$.
- These may be seen in different bases. One important one for our purposes is, extracting around the LCS point $\varphi = 0$, the Frobenius basis:

$$\varpi = \begin{pmatrix} f \\ f^i + fl^i \\ \frac{1}{2!}Y_{ijk}(f^{ij} + 2f^j l^k + fl^j l^k) \\ \frac{1}{3!}Y_{ijk}(f^{ijk} + 3f^{ij}l^k + 3f^i l^j l^k + fl^i l^j l^k) \end{pmatrix}$$

- Defining $(\vartheta_0, \vartheta_i, \vartheta^i, \vartheta^0) = \left(1, \theta_i, \hat{Y}^{ijk}\theta_j\theta_k, \frac{\hat{Y}^{ijk}}{m}\theta_i\theta_j\theta_k\right)$, we then have

$$E(\varphi)_a^b = \begin{pmatrix} \vartheta_a\varpi^b & \vartheta^a\varpi^b \\ \vartheta_a\varpi_b & \vartheta^a\varpi_b \end{pmatrix}$$

- It turns out $U_p(\varphi) = \widetilde{E}(\varphi^p)^{-1} U_p(0) \widetilde{E}(\varphi)$, where $\widetilde{E}(\varphi) \coloneqq E(\varphi)|_{\log(\varphi_i)\to 0}$.
  - We don't have to deal with log terms!
- Advantages:
  - Tackles the whole family of CY3s
  - Relatively quick to evaluate
  - Can be evaluated at conifold loci
  - Convergence in p-adics
- Disadvantages:
  - Can't reach as high of primes as e.g. controlled reduction (see for instance https://github.com/edgarcosta/pycontrolledreduction)

- It turns out $U_p(\varphi) = \widetilde{E}(\varphi^p)^{-1} U_p(0) \widetilde{E}(\varphi)$, where $\widetilde{E}(\varphi) \coloneqq E(\varphi)|_{\log(\varphi_i)\to 0}$.
  - We don't have to deal with log terms!
- Advantages:
  - Tackles the whole family of CY3s
  - Relatively quick to evaluate
  - Can be evaluated at conifold loci
  - Convergence in p-adics
- Disadvantages:
  - Can't reach as high of primes as e.g. controlled reduction (see for instance https://github.com/edgarcosta/pycontrolledreduction)

- It turns out $U_p(\varphi) = \widetilde{E}(\varphi^p)^{-1} U_p(0) \widetilde{E}(\varphi)$, where $\widetilde{E}(\varphi) \coloneqq E(\varphi)|_{\log(\varphi_i) \to 0}$.
    - We don't have to deal with log terms!
- Advantages:
    - Tackles the whole family of CY3s
    - Relatively quick to evaluate
    - Can be evaluated at conifold loci
    - Convergence in p-adics
- Disadvantages:
    - Can't reach as high of primes as e.g. controlled reduction (see for instance https://github.com/edgarcosta/pycontrolledreduction)

Method 1: Picard-Fuchs Equation

Given the Picard-Fuchs equation, create power series ansätze for the periods and higher periods in terms of the complex structure parameters; e.g. $f = \sum_{n=0}^{\infty} c_n \varphi^n$. As we are expanding around $\varphi = 0$, set $c_0 = 1$ as an initial condition. Apply the differential operators to the ansatz, and obtain recurrence relations.

This of course generalizes to higher parameter families.

Example

Given a one-parameter differential operator:

$$\mathcal{L} = \sum_{k=0}^{d} P_k(\varphi)\theta^k,$$

with $P_k(\varphi) \in \mathbb{Z}[\varphi]$ integral polynomials and $\theta = \varphi\partial_\varphi$, then we know $\theta\varphi^n = n\varphi^n$. Apply the operator to the ansatz, find the recurrence, and use the initial condition to solve for these coefficients c_n.

Example

Consider the Picard-Fuchs for the quintic:

$$\begin{aligned}\mathcal{L} &= \theta^4 - 5\varphi(5\theta+1)(5\theta+2)(5\theta+3)(5\theta+4) \\ &= \theta^4 - 3125\varphi\theta^4 - 6250\varphi\theta^3 - 4375\varphi\theta^2 - 1250\varphi\theta - 120\varphi\end{aligned}$$

Apply this to $f = \sum_{n=0}^{\infty} c_n \varphi^n$:

$$\begin{aligned}\mathcal{L}f &= \sum_{n=0}^{\infty} n^4 c(n)\varphi^n - 3125\sum_{n=0}^{\infty} n^4 c(n)\varphi^{n+1} + \cdots \\ &= 0.\end{aligned}$$

This gives a recurrence between $c(n)$ and $c(n+1)$.

- We note that for the higher periods, as they also satisfy the Picard-Fuchs system, we can find them via standard methods to solve differential equations. In our case, we use the Frobenius method.
  - We have $f(z) = \sum c(n)\varphi^n$.
  - To find further solutions from this initial solution, take $f(z,\rho) = \sum c(n+\rho)\varphi^{n+\rho}$. Then we can look at $\partial_\rho f|_{\rho=0}$.
  - This strategy gives us one way to compute the coefficients f^i, f^{ij}, and f^{ijk} from before!

Computing Zeta Functions

Periods: Computations

In [267…

```
L_quintic = theta**4-5*z*(5*theta+1)*(5*theta+2)*(5*theta+3)*(5*theta+4)
kappa = 5
max_n = 1000

periods_quintic = pf_speedrun(L_quintic,kappa,max_n)
```

The operator

$$\theta^4 - 5z(5\theta+1)(5\theta+2)(5\theta+3)(5\theta+4)$$

Satisfies the recurrence relation

$$-3125n^4c(n) - 6250n^3c(n) - 4375n^2c(n) - 1250nc(n) + (n+1)^4c(n+1) - 120c(n)$$

```
It took 2.4455952644348145 seconds to compute 1000 terms of the solution.
c(0) = 1
c(1) = 120
c(2) = 113400
c(3) = 168168000
c(4) = 305540235000
c(5) = 623360743125120
c(6) = 1370874167589326400
c(7) = 3177459078523411968000
c(8) = 7656714453153197981835000
c(9) = 19010638202652030712978200000
d(0) = 0
d(1) = 770
d(2) = 810225
d(3) = 3745679000/3
d(4) = 4627120640625/2
d(5) = 4776890809748904
d(6) = 10589914735183563780
d(7) = 24687653993108095017600
d(8) = 238994525146844285287808625/4
d(9) = 1339662446153766674378966491250/9
d_2(0) = 0
d_2(1) = 2875
d_2(2) = 21040875/4
d_2(3) = 84822610000/9
d_2(4) = 900108233890625/48
d_2(5) = 40360182167406990
d_2(6) = 91943921656621615700
d_2(7) = 10706723300003047945287000/49
d_2(8) = 1682274029589364861124198026875/3136
d_2(9) = 6129440113413577251707286214540625/4536
d_3(0) = 0
d_3(1) = -5750
d_3(2) = -16491875/4
d_3(3) = -233308099375/54
d_3(4) = -1626647874546875/288
d_3(5) = -67140623450467993/8
d_3(6) = -5803135656533650259615/432
d_3(7) = -68839905199851646538471800/3087
```

Method 2: Toric Data

Given the GLSM data/Mori vectors, one can construct the fundamental period [Hosono, Klemm, Theisen, Yau hep-th/9406055v2]. We can do this with `SymPy` and take the appropriate derivatives to find the higher periods.

Imports

[1]:
```
#pip install symengine
```

[2]:
```
#pip uninstall symengine --yes
```

[3]:
```
#pip uninstall sympy --yes
```

[4]:
```
#pip install sympy==1.11.1
```

[5]:
```
import generate_periods_updated_sp as generate_periods
from cytools import Polytope, fetch_polytopes
import flint
import numpy as np
import symengine as se
import sympy as sp
import time
```

Quintic

Setup

```
[6]: g = fetch_polytopes(h11=1,lattice="N",as_list=True)
```

```
[7]: poly = g[1]
     poly.vertices() # picking mirror quintic
```

```
[7]: array([[-1, -1, -1, -1],
            [ 0,  0,  0,  1],
            [ 0,  0,  1,  0],
            [ 0,  1,  0,  0],
            [ 1,  0,  0,  0]])
```

```
[8]: t = poly.triangulate()
     cy = t.get_cy()
```

```
[9]: print(cy.h11())
     print(cy.h21())
```

```
1
101
```

```
[10]: intnums = cy.intersection_numbers(in_basis=True, format='dense').tolist()
      print(intnums)
```

```
[[[5]]]
```

```
[11]: glsm_charge = cy.glsm_charge_matrix(include_origin=True).tolist()
      print(glsm_charge)
```

```
[[-5, 1, 1, 1, 1, 1]]
```

We can then (in a misleading way) compute the periods:

Scratch Work with SymPy

[12]:
```
start_time = time.time()
f0, n_vars = generate_periods.FunPeriod_se(glsm_charge)
f1,f2,f3 = generate_periods.get_higher_periods(f0,n_vars)
end_time = time.time()
print(f'Computing the periods takes {end_time-start_time} seconds')
```

```
Computing the periods takes 0.0021300315856933594 seconds
```

[13]:
```
f0._sympy_().rewrite(sp.harmonic)
```

[13]: $\frac{\Gamma(5n_1+1)}{\Gamma^5(n_1+1)}$

[14]:
```
f1[0]._sympy_()
```

[14]: $-\frac{5\Gamma(5n_1+1)\operatorname{polygamma}(0,n_1+1)}{\Gamma^5(n_1+1)}+\frac{5\Gamma(5n_1+1)\operatorname{polygamma}(0,5n_1+1)}{\Gamma^5(n_1+1)}$

[15]:
```
f2[0][0]._sympy_()
```

[15]: $$\frac{25\Gamma(5n_1+1)\operatorname{polygamma}^2(0,n_1+1)}{\Gamma^5(n_1+1)}-\frac{50\Gamma(5n_1+1)\operatorname{polygamma}(0,n_1+1)\operatorname{polygamma}(0,5n_1+1)}{\Gamma^5(n_1+1)}+\frac{25\Gamma(5n_1+1)\operatorname{polygamma}^2(0,5n_1+1)}{\Gamma^5(n_1+1)}$$
$$-\frac{5\Gamma(5n_1+1)\operatorname{polygamma}(1,n_1+1)}{\Gamma^5(n_1+1)}+\frac{25\Gamma(5n_1+1)\operatorname{polygamma}(1,5n_1+1)}{\Gamma^5(n_1+1)}$$

[16]:
```
f3[0][0][0]._sympy_()
```

[16]: $$-\frac{125\Gamma(5n_1+1)\operatorname{polygamma}^3(0,n_1+1)}{\Gamma^5(n_1+1)}+\frac{375\Gamma(5n_1+1)\operatorname{polygamma}^2(0,n_1+1)\operatorname{polygamma}(0,5n_1+1)}{\Gamma^5(n_1+1)}$$
$$-\frac{375\Gamma(5n_1+1)\operatorname{polygamma}(0,n_1+1)\operatorname{polygamma}^2(0,5n_1+1)}{\Gamma^5(n_1+1)}+\frac{75\Gamma(5n_1+1)\operatorname{polygamma}(0,n_1+1)\operatorname{polygamma}(1,n_1+1)}{\Gamma^5(n_1+1)}$$
$$-\frac{375\Gamma(5n_1+1)\operatorname{polygamma}(0,n_1+1)\operatorname{polygamma}(1,5n_1+1)}{\Gamma^5(n_1+1)}+\frac{125\Gamma(5n_1+1)\operatorname{polygamma}^3(0,5n_1+1)}{\Gamma^5(n_1+1)}$$
$$-\frac{75\Gamma(5n_1+1)\operatorname{polygamma}(0,5n_1+1)\operatorname{polygamma}(1,n_1+1)}{\Gamma^5(n_1+1)}+\frac{375\Gamma(5n_1+1)\operatorname{polygamma}(0,5n_1+1)\operatorname{polygamma}(1,5n_1+1)}{\Gamma^5(n_1+1)}$$

Computing Zeta Functions

Periods: Computations

```
[21]: order = 1000
      start_time = time.time()
      f0,f1,f2,f3 = generate_periods.get_rational_coefficients(cy,order=order,as_harmonic=False,verbosity=0)
      end_time = time.time()
      print(f'Total time: {end_time-start_time} seconds to go to order={order}') # can already see it is slow with the quintic.

      Please have SymEngine and SymPy v1.11.1 installed.
      Total time: 12.733577251434326 seconds to go to order=1000

[22]: f0[:10],f1[0][:10],f2[0][0][:10],f3[0][0][0][:10]

[22]: ([1,
        120,
        113400,
        168168000,
        305540235000,
        623360743125120,
        1370874167589326400,
        3177459078523411968000,
        7656714453153197981835000,
        19010638202652030712978200000],
       [0,
        770,
        810225,
        3745679000/3,
        4627120640625/2,
        4776890809748904,
        10589914735183563780,
        24687653993108095017600,
        238994525146844285287808625/4,
        133966244615376667437896649l250/9],
       [0,
        1150,
        4208175/2,
        33929044000/9,
        180021646778125/24,
        16144072866962796,
        36777568662648646280,
        4282689320001219178114800/49,
        33645480591787297222483960537 5/1568,
        122588802268271545034145724 2908125/2268],
       [0,
        -6900,
```

- This is slow*, even compared to what can be done in `Mathematica`!
  - Note, that this is already *after* having made a number of optimizations. Namely, we use an old version of `SymPy` (1.11.1) which allows us to cache the evaluations of these polygammas automatically, and we use `SymEngine`, a backend written in `C`. `SymPy` 1.11.1 by itself took $\mathcal{O}(20s)$, and 1.13.3 by itself took more than 10min before I aborted the evaluation. 1.13.3 and `SymEngine` brought the time to $\mathcal{O}(480s)$.
- However, as an upside, we don't need the Picard-Fuchs equation to compute the periods in this way.

- This is slow*, even compared to what can be done in `Mathematica`!
    - Note, that this is already *after* having made a number of optimizations. Namely, we use an old version of `SymPy` (1.11.1) which allows us to cache the evaluations of these polygammas automatically, and we use `SymEngine`, a backend written in `C`. `SymPy` 1.11.1 by itself took $\mathcal{O}(20s)$, and 1.13.3 by itself took more than 10min before I aborted the evaluation. 1.13.3 and `SymEngine` brought the time to $\mathcal{O}(480s)$.
- However, as an upside, we don't need the Picard-Fuchs equation to compute the periods in this way.

Knowing the periods is also needed for countless other applications:

- Mirror map: $t^i = \frac{1}{2\pi i}\frac{\varpi^i}{\omega^0} \sim \log(z^i) + \mathcal{O}(z)$
  - $\varpi^i \sim \log(z_i) + \mathcal{O}(z)$
  - $\varpi^0 \sim 1 + \mathcal{O}(z)$
- GVs/GWs [cygv (Andres Rios Tascon), 2303.00757]
- Moduli stabilization

Knowing the periods is also needed for countless other applications:

- Mirror map: $t^i = \frac{1}{2\pi i}\frac{\varpi^i}{\omega^0} \sim \log(z^i) + \mathcal{O}(z)$
  - $\varpi^i \sim \log(z_i) + \mathcal{O}(z)$
  - $\varpi^0 \sim 1 + \mathcal{O}(z)$
- GVs/GWs [`cygv` (Andres Rios Tascon), 2303.00757]
- Moduli stabilization

Knowing the periods is also needed for countless other applications:

- Mirror map: $t^i = \frac{1}{2\pi i}\frac{\varpi^i}{\omega^0} \sim \log(z^i) + \mathcal{O}(z)$
  - $\varpi^i \sim \log(z_i) + \mathcal{O}(z)$
  - $\varpi^0 \sim 1 + \mathcal{O}(z)$
- GVs/GWs [`cygv` (Andres Rios Tascon), 2303.00757]
- Moduli stabilization

Implementation

We have implemented* both methods in Python, and are preparing to release the code as `ToricZeta`.

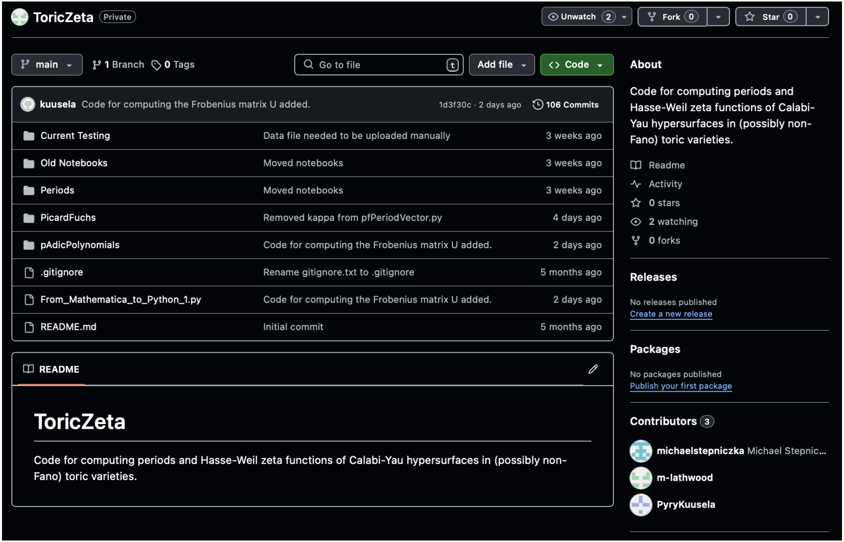
- Method 1: PFs
  - Plus: Fast; can handle multi-parameter families
- Method 2: Directly in SymPy/SymEngine
  - Plus: Don't need PF
  - Minus: Slow... need to do testing on 2-parameter families to see if it can ever work

Implementation

We have implemented* both methods in Python, and are preparing to release the code as `ToricZeta`.

- Method 1: PFs
  - Plus: Fast; can handle multi-parameter families
- Method 2: Directly in SymPy/SymEngine
  - Plus: Don't need PF
  - Minus: Slow... need to do testing on 2-parameter families to see if it can ever work

Implementation



There is some key benchmarking against `CY3Zeta`:

- Polynomial multiplication in `Mathematica` is slow:

```
In[1]:= Clear[a]
        a[n_] := a[n] = (5*(5n-1)*(5n-2)*(5n-3)*(5n-4))/n^4 * a[n-1]
        a[0] = 1;

        Clear[b]
        b[n_] :=
         b[n] = -4/n * a[n] + (1250*(2n-1)*(5n^2-5n+1))/n^4 * a[n-1] +
           (5*(5n-1)*(5n-2)*(5n-3)*(5n-4))/n^4 * b[n-1]
        b[0] = 0;

In[7]:= poly1 = Sum[a[i]*φ^i, {i, 0, 5000}];
        poly2 = Sum[b[i]*φ^i, {i, 0, 5000}];

In[9]:= Timing[poly1*poly2;]
Out[9]= {0.000392, Null}

In[10]:= ser1 = Sum[a[i]*φ^i, {i, 0, 1000}] + O[φ]^1001;
         ser2 = Sum[b[i]*φ^i, {i, 0, 1000}] + O[φ]^1001;

In[12]:= Timing[ser1*ser2;]
Out[12]= {1.61314, Null}
```

Implementation

Python Multiplication

Instead, we make use of `FLINT` (Fast Library for Number Theory) in Python:

fmpq polynomials

[2]:
```
def period_coeffs_iterative_flint(n):
    fund_period_coeffs = [fmpq(1,1)]
    for i in range(1, n):
        fund_period_coeffs.append(fmpq(5*(5*i-1)*(5*i-2)*(5*i-3)*(5*i-4),i**4)*fund_period_coeffs[-1])
    fund = fund_period_coeffs

    log_period_coeffs = [fmpq(0,1)]
    for i in range(1, n):
        log_period_coeffs.append(fmpq(-4,i)*fund[i]+fmpq(1250*(2*i-1)*(5*i**2-5*i+1),i**4)*fund[i-1]+fmpq(5*(5*i-1)*(5*i-2)*(5*i-3)*(5*
    logcoeffs = log_period_coeffs

    return [fund,logcoeffs]
```

[3]:
```
n = 1000
periods_flint = period_coeffs_iterative_flint(n)
p1 = fmpq_poly(periods_flint[0])
p2 = fmpq_poly(periods_flint[1])
```

[4]:
```
start_time = time.time()
flint_mult = p1*p2
end_time = time.time()
execution_time = end_time - start_time
print(execution_time)
```

```
0.09912896156311035
```

- `FLINT` can't be used just as-is for multiparameter cases... need to `pip install python-flint==0.7.0a5` to have access to multivariate polynomials
- We needed to implement truncated polynomial multiplication
- We needed to implement Laurent series by hand (as we have rational functions in our U matrix, but currently only polynomials are supported)

- `FLINT` can't be used just as-is for multiparameter cases... need to `pip install python-flint==0.7.0a5` to have access to multivariate polynomials
- We needed to implement truncated polynomial multiplication
- We needed to implement Laurent series by hand (as we have rational functions in our U matrix, but currently only polynomials are supported)

- `FLINT` can't be used just as-is for multiparameter cases... need to `pip install python-flint==0.7.0a5` to have access to multivariate polynomials
- We needed to implement truncated polynomial multiplication
- We needed to implement Laurent series by hand (as we have rational functions in our U matrix, but currently only polynomials are supported)

`CY3Zeta`

- Intersection numbers, inverse intersection numbers
- Conifold locus
- Other singular loci
- Period coefficients

`ToricZeta`

- (In theory) `CYTools` CalabiYau object
- (In practice) PF equation

Conclusions

- We have implemented the Dwork deformation method in Python, with significant speed improvements over the Mathematica implementation
- In particular, this Python implementation is built to be compatible with e.g. `CYTools`, giving us access to the largest collection of currently-known CY3s
- We will then (given the PF equation) be able find zeta functions of these families

Conclusions

- We have implemented the Dwork deformation method in Python, with significant speed improvements over the Mathematica implementation
- In particular, this Python implementation is built to be compatible with e.g. `CYTools`, giving us access to the largest collection of currently-known CY3s
- We will then (given the PF equation) be able find zeta functions of these families

Conclusions

- We have implemented the Dwork deformation method in Python, with significant speed improvements over the Mathematica implementation
- In particular, this Python implementation is built to be compatible with e.g. `CYTools`, giving us access to the largest collection of currently-known CY3s
- We will then (given the PF equation) be able find zeta functions of these families

Future Work

- As the controlled reduction method is suitable for calculating zeta functions of specific points in moduli space, we can use `ToricZeta` at low primes to find interesing points!
- It would be awesome to automate the finding of these Picard-Fuchs equations via Griffiths-Dwork reduction
- These methods can also be implemented for higher Calabi-Yau n-folds

Future Work

- As the controlled reduction method is suitable for calculating zeta functions of specific points in moduli space, we can use `ToricZeta` at low primes to find interesing points!
- It would be awesome to automate the finding of these Picard-Fuchs equations via Griffiths-Dwork reduction
- These methods can also be implemented for higher Calabi-Yau n-folds

Future Work

- As the controlled reduction method is suitable for calculating zeta functions of specific points in moduli space, we can use `ToricZeta` at low primes to find interesing points!
- It would be awesome to automate the finding of these Picard-Fuchs equations via Griffiths-Dwork reduction
- These methods can also be implemented for higher Calabi-Yau n-folds

Thank you for listening!