

Computing Grothendieck Point Residues via Solving Holonomic Systems of First Order Partial Differential Equations II

Shinichi Tajima (Osaka Metropolitan University),

Katsusuke Nabeshima (Tokyo University of Science)

nabeshima@rs.tus.ac.jp

ISSAC 2026
Oldenburg, Germany

2026, July, 15

- Grothendieck point residues play an important role in complex analysis, algebraic geometry, singularity theory, and foliation theory.
- They can be used to compute Futaki invariants and Camacho-Sad-Suwa indices of singular curves.
- This motivates the development of efficient algorithms for computing Grothendieck point residues.

This work continues our ISSAC 2021 paper, in which we presented an algorithm for computing Grothendieck point residues by solving a holonomic system of first-order partial differential equations.

S.Tajima and K. Nabeshima:

Computing Grothendieck point residues via solving holonomic systems of first order linear partial differential equations.
Proc. ISSAC 2021, pp.361-368, ACM.

Today's talk

Today we present an outline of new methods for computing Grothendieck point residues $\left(\frac{1}{2\pi\sqrt{-1}}\right)^m \int \cdots \int_{\gamma_\epsilon} \frac{h(z)dz}{f_1(z)f_2(z)\cdots f_m(z)}$ by solving holonomic systems of first order linear partial differential equations.

Setting (local cohomology)

K : the field $\mathbb{Q}$ of rational numbers,

$K[x] := K[x_1, \dots, x_m]$,

$f_1(x), \dots, f_m(x) \in K[x]$: regular sequence of polynomials

$J = \langle f_1(x), \dots, f_m(x) \rangle$: ideal

Assume that the common locus $Z = \{x \in \mathbb{C}^m \mid f_1(x) = \dots = f_m(x)\}$ is a zero-dimensional variety. Assume also that there is an open neighborhood $X \subset \mathbb{C}^m$ of the origin $O \in \mathbb{C}^m$ such that $Z \cap X = \{O\}$.

Let $H_{[Z]}^m(K[x])$ denote the highest local cohomology supported on Z defined as

$$H_{[Z]}^m(K[x]) = \lim_{k \rightarrow \infty} \operatorname{Ext}_{K[x]}^m(K[x]/(\sqrt{J})^k, K[x]).$$

Let I be the primary component of J such that $\sqrt{I} = \mathfrak{m}$, where $\mathfrak{m} = \langle x_1, x_2, \dots, x_m \rangle$ is the maximal ideal in $K[x]$.

Then the ideal J can be decomposed as $J = I \cap J'$, where $J' = J : \mathfrak{m}^\infty$ is the saturation of J with respect to $\mathfrak{m}$.

Set

$$H_{[O]}^m(K[x]) = \lim_{k \rightarrow \infty} \operatorname{Ext}_{K[x]}^m(K[x]/\mathfrak{m}^k, K[x]),$$
$$H_{[Z']}^m(K[x]) = \lim_{k \rightarrow \infty} \operatorname{Ext}_{K[x]}^m(K[x]/(\sqrt{J'})^k, K[x]),$$

where $Z' = Z - \{O\}$. Then, we have

$$H_{[Z]}^m(K[x]) = H_{[O]}^m(K[x]) \oplus H_{[Z']}^m(K[x]).$$

Grothendieck point residue

$h(x) \in K[x]$: holomorphic function at O .

For $h(x)$, the **Grothendieck point residue** at the origin O , denoted by

$$\text{res}_{\{O\}}(h(x), \left[\begin{array}{c} 1 \\ f_1 f_2 \cdots f_m \end{array} \right]),$$

of the differential form $\frac{h(x)dx}{f_1(x)f_2(x)\cdots f_m(x)}$ can be expressed as the Weil integral

$$\left(\frac{1}{2\pi\sqrt{-1}} \right)^m \int \cdots \int_{\gamma_\epsilon} \frac{h(x)}{f_1(x)f_2(x)\cdots f_m(x)} dx,$$

where $dx = dx_1 \wedge dx_2 \wedge \cdots \wedge dx_m$, and where γ_ϵ is a real m -dimensional cycle:

$\gamma_\epsilon = \{x \in X \mid |f_1(x)| = |f_2(x)| = \cdots = |f_m(x)| = \epsilon\}$, with $0 < \epsilon \ll 1$.

$\left[\begin{array}{c} 1 \\ f_1 f_2 \cdots f_m \end{array} \right] \in H_{[Z]}^m(K[x])$ where $[]$ is the Grothendieck symbol.

Some facts

$$h = \sum \mathbf{c}_\alpha x^\alpha \in K[x] \text{ where } c_\alpha \in K \text{ and } x^\alpha = x_1^{\alpha_1} \cdots x_m^{\alpha_m}. \quad \left[\begin{array}{c} 1 \\ x^\kappa \end{array} \right] \in H_{[0]}^m(K[x])$$

$$\text{res}_{\{O\}}(h, \left[\begin{array}{c} 1 \\ x^\kappa \end{array} \right]) = \mathbf{c}_{\kappa-1}$$

where $\kappa = (k_1, \dots, k_m) \in \mathbb{Z}_{\geq 1}^m$ and $\kappa - 1 = (k_1 - 1, \dots, k_m - 1)$.

$$\text{res}_{\{O\}}(h, \sum b_\alpha \left[\begin{array}{c} 1 \\ x^\alpha \end{array} \right]) = \sum b_\alpha \text{res}_{\{O\}}(h, \left[\begin{array}{c} 1 \\ x^\alpha \end{array} \right])$$

Example

$$h_1 = \sum c_{(i,j)} x_1^i x_2^j$$

$$\begin{aligned} \text{res}_{\{O\}}(h_1, \left[\begin{array}{c} 1 \\ x_1 x_2^6 \end{array} \right] - \frac{1}{3} \left[\begin{array}{c} 1 \\ x_1^3 x_2 \end{array} \right]) &= \text{res}_{\{O\}}(h_1, \left[\begin{array}{c} 1 \\ x_1 x_2^6 \end{array} \right]) - \frac{1}{3} \text{res}_{\{O\}}(h_1, \left[\begin{array}{c} 1 \\ x_1^3 x_2 \end{array} \right]) \\ &= c_{(0,5)} - \frac{1}{3} c_{(2,0)} \end{aligned}$$

- We would like to compute $\text{res}_{\{O\}}(h(x), \sigma_J)$ where $h(x) \in K[x]$ and $\sigma_J = \begin{bmatrix} 1 \\ f_1 f_2 \cdots f_m \end{bmatrix} \in H_{[Z]}^m(K[x])$.
- If $\begin{bmatrix} 1 \\ f_1 f_2 \cdots f_m \end{bmatrix}$ can be transformed into the form $\sum_{\kappa \in \mathbb{N}^m} c_\kappa \begin{bmatrix} 1 \\ x^\kappa \end{bmatrix} \in H_{[O]}^m(K[x])$, then the Grothendieck point residue can be computed easily.
- This transformation is called the **transformation law**.
- The transformation law is described theoretically in [Hartshorne's book "Residues and Duality" \(1966\)](#), but no computational method is provided there.

We presented an algorithm for computing the transformation law and, consequently, Grothendieck point residues in the following paper.

[S. Tajima and K. Nabeshima: An effective method for computing Grothendieck point residue mappings, J. Algebra, **593**, 568--588, 2022](#)

However, the method based on the transformation law is not computationally efficient. Therefore, in our ISSAC 2021 paper, we proposed an alternative method based on first-order partial differential equations.

Our result of ISSAC2021

$\sigma_J = \begin{bmatrix} 1 \\ f_1 f_2 \cdots f_m \end{bmatrix} \in H_{[Z]}^m(K[x])$ can be decomposed as

$$\sigma_J = \tau_J + \sigma'_J, \quad \tau_J \in H_{[O]}^m(K[x]), \quad \sigma'_J \in H_{[Z']}^m(K[x]). \quad (Z' = Z - \{O\})$$

Theorem 1 (Tajima-Nabeshima, ISSAC2021)

$$\text{res}_{\{O\}}(h, \sigma_J) = \text{res}_{\{O\}}(h, \tau_J).$$

How to compute τ_J ?

We can use partial differential operators to compute τ_J .

$D := K[x_1, \dots, x_n, \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n}]$: ring of partial diff. op. (Weyl algebra),
 $\text{Ann}_D(\sigma_J) = \{P \in D \mid P\sigma_J = 0\}$

Lemma 1 (Tajima-Nabeshima, ISSAC2021)

For any $P \in \text{Ann}_D(\sigma_J)$, $P\tau_J = 0$.

Lemma 2 (Tajima-Nabeshima, ISSAC2021)

$$\{\psi \in H_{[0]}^m(K[x]) \mid P\psi = 0, \forall P \in \text{Ann}_D(\sigma_J)\} = \text{Span}_K(\tau_J).$$

Partial differential operators in $\text{Ann}_D(\sigma_J)$ enable us to compute τ_J where $\sigma_J = \tau_J + \sigma'_J$.

Theorem 2 (Tajima-Nabeshima, ISSAC2021)

There exist 1st order partial diff. operators in $\text{Ann}_D(\sigma_J)$ that characterize the vector space $\{\psi \in H_{[0]}^m(K[x]) \mid P\psi = 0, \forall P \in \text{Ann}_D(\sigma_J)\}$.

Computational method presented in ISSAC 2021

Step 1: Compute a set T_P of 1st order partial differential operators in $\text{Ann}_D(\sigma_J)$.

Step 2: Compute a basis Ψ of the vector space $\{\psi \in H_{[0]}^m(K[x]) \mid P\psi = 0, \forall P \in T_P\}$.
(There exists an algorithm for computing Ψ .)

Step 3: Compute τ_J using Ψ .
(There exists an algorithm for computing τ_J from Ψ .)

Algorithm 1 (ISSAC 2021)

Let $\mathbf{e}_i$ be the transpose of $(0 \cdots 0 \mathbf{1} 0 \cdots 0)$ and $\frac{\partial \varphi}{\partial \mathbf{x}_i}$ ($i = 1, 2, \dots, m$) the transpose of $(\frac{\partial f_1}{\partial \mathbf{x}_i}, \frac{\partial f_2}{\partial \mathbf{x}_i}, \dots, \frac{\partial f_m}{\partial \mathbf{x}_i})$.

Algorithm 1 (ISSAC2021)

Input: $f_1, \dots, f_m$: a regular sequence.

Output T_P : generators of first order annihilators of $\sigma_J \in H_{[Z]}^m(K[x])$.

Step 1: Compute a Gröbner basis Syz of the syzygy module of

$$\frac{\partial \varphi}{\partial \mathbf{x}_1}, \frac{\partial \varphi}{\partial \mathbf{x}_2}, \dots, \frac{\partial \varphi}{\partial \mathbf{x}_m}, f_1 \mathbf{e}_1, f_2 \mathbf{e}_1, \dots, f_m \mathbf{e}_1, f_1 \mathbf{e}_2, f_2 \mathbf{e}_2, \\ f_3 \mathbf{e}_2, \dots, f_m \mathbf{e}_2, f_1 \mathbf{e}_3, \dots, f_m \mathbf{e}_m;$$

Step 2: $T_P \leftarrow \emptyset$;

For each $(p_1, p_2, \dots, p_m, q_{1,1}, q_{1,2}, \dots, q_{1,m}, q_{2,1}, q_{2,2}, \dots, q_{2,m}, \\ q_{3,1}, \dots, q_{m,m}) \in \text{Syz}$;

$$v \leftarrow p_1(x) \frac{\partial}{\partial x_1} + p_2(x) \frac{\partial}{\partial x_2} + \cdots + p_m(x) \frac{\partial}{\partial x_m};$$

$$p_0 \leftarrow -(q_{1,1} + q_{2,2} + \cdots + q_{m,m}); \quad T_P \leftarrow T_P \cup \{v + p_0\};$$

Return T_P ;

New results (ISSAC2026)

$$(\sigma_J = \tau_J + \sigma'_J)$$

Lemma 3

$$\{\psi \in H_{[0]}^m(K[x]) \mid P\psi = 0, \forall P \in \text{Ann}_D(\tau_J)\} = \text{Span}_K(\tau_J).$$

Lemma 4

There exist 1st order partial diff. operators in $\text{Ann}_D(\tau_J)$ that characterize the vector space $\{\psi \in H_{[0]}^m(K[x]) \mid P\psi = 0, \forall P \in \text{Ann}_D(\tau_J)\}$.

New computational method

Step 1: Compute a set T_P of 1st order partial differential operators in $\text{Ann}_D(\tau_J)$.

Step 2: Compute a basis Ψ of the vector space $\{\psi \in H_{[0]}^m(K[x]) \mid P\psi = 0, \forall P \in T_P\}$.
(There exists an algorithm for computing Ψ .)

Step 3: Compute τ_J using Ψ .
(There exists an algorithm for computing τ_J from Ψ .)

Algorithm 2

Algorithm 2 (new result)

Input: $f_1, \dots, f_m$: a regular sequence.

Output T_P : generators of first order annihilators of $\tau_J \in H_{[\circ]}^m(K[x])$.

Step 1: Compute $g(x) \in J : \mathfrak{m}^\infty$ s.t. $g(0) \neq 0$;

Step 2: Compute a Gröbner basis Syz of the syzygy module of $g \frac{\partial \varphi}{\partial x_1}, g \frac{\partial \varphi}{\partial x_2}, \dots, g \frac{\partial \varphi}{\partial x_m}, f_1 \mathbf{e}_1, f_2 \mathbf{e}_1, \dots, f_m \mathbf{e}_1, f_1 \mathbf{e}_2, f_2 \mathbf{e}_2, f_3 \mathbf{e}_2, \dots, f_m \mathbf{e}_2, f_1 \mathbf{e}_3, \dots, f_m \mathbf{e}_m$;

Step 3: $T_P \leftarrow \emptyset$;

For each $(p_1, p_2, \dots, p_m, q_{1,1}, q_{1,2}, \dots, q_{1,m}, q_{2,1}, q_{2,2}, \dots, q_{2,m}, q_{3,1}, \dots, q_{m,m}) \in \text{Syz}$;

$$v \leftarrow p_1(x) \frac{\partial}{\partial x_1} + p_2(x) \frac{\partial}{\partial x_2} + \dots + p_m(x) \frac{\partial}{\partial x_m};$$

$$p_0 \leftarrow -(q_{1,1} + q_{2,2} + \dots + q_{m,m}); \quad T_P \leftarrow T_P \cup \{v + p_0\};$$

Return T_P ;

- Algorithm 1 : syzygy computation is the bottleneck.
- Algorithm 2 : still requires syzygies.
- **Goal:** eliminate syzygy computation completely.
- **Result:** Algorithm 3 is syzygy-free.

We would like a syzygy-free algorithm.

By combining

- saturation ideals,
- standard bases, and
- the extended Gröbner basis algorithm,

we developed a completely **syzygy-free algorithm** for computing generators of the first-order annihilator of $\tau_J \in H_{[O]}^m(K[x])$ which we call Algorithm 3.

Algorithm 3

The key point is that we replace the syzygy computation by combining saturation ideals, standard bases, and the extended Gröbner basis algorithm.

Algorithm 3

Input: $f_1, \dots, f_m$: a regular sequence.

Output T_P : generators of first order annihilators of τ_J .

Step 1: Compute $g(x) \in J : \mathfrak{m}^\infty$ s.t. $g(0) \neq 0$;

Step 2: Compute a basis Ψ_J of the vector space
 $H_J = \{\psi \in H_{[0]}^n(K[x]) \mid \forall g \in J, g * \psi = 0\}$;

Step 3: If there exists $\varrho_J \in \Psi_J$ such that $K[x] * \varrho_J = H_J$, then goto **Step 4**, if not, execute another algorithm for computing the 1st order partial differential ops;

Step 4: **for each** j **from** 1 **to** m **do**

 Compute $A_j = \text{Ann}_{K[x]}(\text{Span}_{K[x]} \{ \frac{\partial f_i}{\partial x_j} * \varrho_J \mid 1 \leq i \leq m \})$;

 Compute a **standard basis** S_j of A_j and a **monomial basis** M_j of $K[x]/\langle LT(S_j) \rangle$;
 end-for

Step 5: Set $v = \sum_{i=1}^m \left(\sum_{x^\alpha \in M_i} c_{i,\alpha} x^\alpha \right) \frac{\partial}{\partial x_i}$ where $c_{i,\alpha}$ s are indeterminates;

Step 6: $Eq_1 = \{h_\beta = 0 \mid \beta \in \bigcup_{i=1}^m B_i, i = 1, 2, \dots, m\}$ where $v(f_1) * \varrho_J = \sum_{\beta_1 \in B_1} h_{\beta_1} \zeta^{\beta_1}, \dots, v(f_m) * \varrho_J = \sum_{\beta_m \in B_m} h_{\beta_m} \zeta^{\beta_m}$ and $h_{\beta_1}, \dots, h_{\beta_m}$ are linear functions of c_α s;

Step 7: Compute a basis B of the solution space of the system Eq_1 of linear equations;
 $V \leftarrow \emptyset$;
while $B \neq \emptyset$ **do**
 Take b from B ; $B \leftarrow B \setminus \{b\}$; $v_b \leftarrow$ Substitute b into $c_{i,\alpha}$ s of v ;
 $V \leftarrow V \cup \{v_b\}$;
end-while

Step 8: $T_P \leftarrow \emptyset$;
while $V \neq \emptyset$ **do**
 Take v_b from V ; $V \leftarrow V \setminus \{v_b\}$;
for each j **from** 1 **to** m **do**
 Compute $q_{j1}, q_{j2}, \dots, q_{jm} \in K[x]$ such that
 $gv_b(f_j) = q_{j,1}f_1 + q_{j,2}f_2 + \dots + q_{j,j}f_j + \dots + q_{j,m}f_m$;
 (by using the extended Gröbner basis algorithm)
end-for
 $p_0 \leftarrow q_{1,1} + q_{2,2} + \dots + q_{m,m}$; $T_P \leftarrow T_P \cup \{gv_b + p_0\}$;
end-while

Return T_P ;

Thus, Algorithm 3 computes first-order annihilators without any syzygy computation.

Example

$$f = x^3y + x^2y + xy^4.$$

The hypersurface defined by $f = 0$ has the isolated singularity at the origin in $\mathbb{C}^2$.

We would like to compute $\text{res}_{\{O\}}(3x^2 + 4, \left[\begin{smallmatrix} 1 \\ (\frac{\partial f}{\partial x})(\frac{\partial f}{\partial y}) \end{smallmatrix} \right])$.

1st order partial differential operators

- $gv_1 + \frac{198}{25}x^6y + \frac{1886}{25}x^5y + \frac{432}{25}x^4y^4 + \frac{4662}{25}x^4y - \frac{9576}{25}x^3y^4 + 194x^3y + \frac{738}{25}x^2y^7 - \frac{23928}{25}x^2y^4 + \frac{112}{25}x^2y + \frac{276}{5}xy^7 - \frac{18624}{25}xy^4 - \frac{1764}{25}xy + \frac{642}{25}y^7 - \frac{13328}{25}y^4,$
- $gv_2 + \frac{198}{25}x^6 + \frac{1886}{25}x^5 + \frac{432}{25}x^4y^3 + \frac{4662}{25}x^4 - \frac{9576}{25}x^3y^3 + 194x^3 + \frac{738}{25}x^2y^6 - \frac{23928}{25}x^2y^3 + \frac{112}{25}x^2 + \frac{276}{5}xy^6 - \frac{18624}{25}xy^3 - \frac{1764}{25}x + \frac{642}{25}y^6 - \frac{13328}{25}y^3,$
- $gv_3 - 2417x^3y^2 - 5032x^2y^2 + 12xy^5 - 3301xy^2 + 12y^5 - 686y^2,$
- $gv_4 + \frac{792}{25}x^6y + \frac{3144}{25}x^5y + \frac{1728}{25}x^4y^4 + \frac{4848}{25}x^4y + \frac{4896}{25}x^3y^4 - 2705x^3y + \frac{2952}{25}x^2y^7 + \frac{5088}{25}x^2y^4 - \frac{137752}{25}x^2y + \frac{1344}{5}xy^7 - \frac{4596}{25}xy^4 - \frac{85981}{25}xy + \frac{4968}{25}y^7 + \frac{588}{25}y^4 - 686y,$
- $gv_5 + 80x^4y^2 - \frac{7690}{3}x^3y^2 - 164x^2y^5 - \frac{13268}{3}x^2y^2 - 336xy^5 - \frac{6506}{3}xy^2 - 172y^5 - \frac{784}{3}y^2,$
- $gv_6 + \frac{792}{25}x^6 + \frac{3144}{25}x^5 + \frac{1728}{25}x^4y^3 + \frac{4848}{25}x^4 + \frac{4896}{25}x^3y^3 - 2705x^3 + \frac{2952}{25}x^2y^6 + \frac{5088}{25}x^2y^3 - \frac{137752}{25}x^2 + \frac{1344}{5}xy^6 - \frac{4596}{25}xy^3 - \frac{85981}{25}x + \frac{4968}{25}y^6 + \frac{588}{25}y^3 - 686.$

where $v_1 = (\frac{6}{25}y^7)\frac{\partial}{\partial x} + (-y^5)\frac{\partial}{\partial y}$, $v_2 = (\frac{6}{25}y^6)\frac{\partial}{\partial x} + (-y^4)\frac{\partial}{\partial y}$,
 $v_3 = (-3xy^2)\frac{\partial}{\partial x} + (-y^3)\frac{\partial}{\partial y}$, $v_4 = (-3xy + \frac{24}{25}y^7)\frac{\partial}{\partial x} + (-y^2)\frac{\partial}{\partial y}$,
 $v_5 = (-\frac{14}{3}xy^2 - \frac{4}{3}y^5)\frac{\partial}{\partial x} + (-x)\frac{\partial}{\partial y}$, $v_6 = (-3x + \frac{24}{25}y^6)\frac{\partial}{\partial x} + (-y)\frac{\partial}{\partial y}$.

$$\tau_J = -\frac{4}{7} \begin{bmatrix} 1 \\ x^3 y \end{bmatrix} + \frac{1}{7} \begin{bmatrix} 1 \\ x^2 y^4 \end{bmatrix} + \frac{72}{49} \begin{bmatrix} 1 \\ x^2 y \end{bmatrix} - \frac{2}{7} \begin{bmatrix} 1 \\ xy^7 \end{bmatrix} - \frac{11}{49} \begin{bmatrix} 1 \\ xy^4 \end{bmatrix} - \frac{988}{343} \begin{bmatrix} 1 \\ xy \end{bmatrix}.$$

$$\begin{aligned} \operatorname{res}_{\{O\}}(3x^2 + 4, \begin{bmatrix} 1 \\ (\frac{\partial f}{\partial x})(\frac{\partial f}{\partial y}) \end{bmatrix}) &= \operatorname{res}_{\{O\}}(3x^2 + 4, \tau_J) \\ &= 3 \cdot \left(-\frac{4}{7}\right) + 4 \cdot \left(-\frac{988}{343}\right) = -\frac{4540}{343}. \end{aligned}$$

Comparisons

We have implemented the following four algorithms for computing Grothendieck point residues in the computer algebra system Risa/Asir.

trans. law	J. Algebra (2022)
Algorithm 1	presented in ISSAC2021
Algorithm 2	NEW (with syzygy computation)
Algorithm 3	NEW (syzygy-free algorithm)

PC [OS: Windows (64 bit), CPU: Intel Core i9-7900 @ 3.30 GHz, RAM 128 GB]

The following 8 polynomials that define an isolated singularity at the origin are used for the tests to compare the efficiency of four algorithms.

$$\begin{aligned} S_{11} &: x^4 + y^2z + xz^2 + x^3z, & Q_{11} &: x^3 + y^2z + xz^3 + z^5, \\ S_{14} &: x^2z + yz^2 + y^5 + y^3z + z^3, & V'_{18} &: x^3 + y^4 + y^2z^2 + z^4 + xyz^2, \\ S_{23} &: x^2z + yz^2 + y^8 + xy^6, & Q_{23} &: x^3 + yz^2 + xy^7 + y^{11} + y^{12}, \\ S_{24} &: x^2z + yz^2 + xy^6 + y^5z + y^9, & Q_{24} &: x^3 + yz^2 + y^{11} + xy^8 + xy^9, \\ Q_{28} &: x^3 + yz^2 + y^{13} + xy^9 + xy^{10}, & S_{16} &: x^2z + yz^2 + xy^4 + y^6 + z^3, \\ U_{16} &: x^3 + xz^2 + y^5 + y^2z^2 + y^3z^2. \end{aligned}$$

In the table, f_1, f_2 and f_3 mean $f_1 = \frac{\partial f}{\partial x}$, $f_2 = \frac{\partial f}{\partial y}$, $f_3 = \frac{\partial f}{\partial z}$, and the implementations return local cohomology class $\tau_{J(\alpha)}$ associated to the Grothendieck point residue mappings

Timing

	f	$\text{res}_{\{O\}}$	trans. law	Algorithm 1		Algorithm 2		Algorithm 3	
			time	#PD	time	#PD	time	#PD	time
1	S_{11}	$1/(f_1^2 f_2 f_3^4)$	19.95	33	4.828	29	8.906	14	23.66
2	S_{11}	$1/(f_1^2 f_2 f_3^5)$	2936	32	23.52	30	34.06	12	69.08
3	Q_{11}	$1/(f_1^3 f_2 f_3^4)$	36.3	22	10.33	30	33.48	13	69.5
4	Q_{11}	$1/(f_1^4 f_2 f_3^4)$	$> 50m$	24	23.53	30	167.7	14	314.4
5	S_{14}	$1/(f_1 f_2^2 f_3)$	0.6094	21	5.141	19	5.344	13	4.859
6	V'_{18}	$1/(f_1^2 f_2 f_3)$	0.6719	31	3.641	29	4.016	15	2.578
7	V'_{18}	$1/(f_1 f_2 f_3^3)$	463.4	37	93.95	34	91.3	25	61.5
8	S_{23}	$1/(f_1 f_2 f_3)$	0.1563	29	4.595	24	5.078	16	5.688
9	S_{23}	$1/(f_1^2 f_2 f_3)$	11.2	31	217.3	28	132.3	16	108.7
10	Q_{23}	$1/(f_1^2 f_2 f_3)$	11.13	24	115.8	20	57.1	32	15.2
11	Q_{23}	$1/(f_1^2 f_2 f_3^2)$	$> 50m$	30	562	26	1162	11	35.34
12	S_{24}	$1/(f_1 f_2 f_3)$	47.38	43	20.95	34	64.16	18	988
13	Q_{24}	$1/(f_1^3 f_2 f_3)$	22.86	22	335.5	20	171.3	6	13.13
14	Q_{28}	$1/(f_1^3 f_2 f_3)$	878.2	25	1579	20	716.8	6	37.92
15	S_{16}	$1/(f_1 f_2 f_3)$	2.219	31	1.656	23	5.969	13	6.063
16	S_{16}	$1/(f_1^2 f_2 f_3)$	$> 50m$	33	40.42	26	259.3	13	40.08
17	U_{16}	$1/(f_1^2 f_2 f_3^3)$	$> 50m$	44	525.8	40	297.1	19	213.8
18	U_{16}	$1/(f_1^3 f_2 f_3^3)$	$> 50m$	-	$> 50m$	-	$> 50m$	21	1492

The time is given in second (CPU time), “ $> 50m$ ” means it takes more than 50 minutes, and “#PD” means a number of 1st order partial differential operators that are obtained in the computation.

- Algorithm 3 is not always the fastest. However, on computationally difficult examples, it often provides substantial performance improvements.

Today I introduced two new algorithms for computing Grothendieck point residues. The main contribution is Algorithm 3, which computes first-order annihilators without any syzygy computation.

Thank you for your attention.