Refined Algorithms to Compute Syzygies

Andreas Steenpaß

TU Kaiserslautern

Algebraic Geometry and its Applications
Lahore, August 2018

Outline

Syzygies

Setup:

- Let K a field.
- Set $R := K[x_1, \dots, x_n]$.

Syzygies

Setup:

- Let K a field.
- Set $R := K[x_1, \ldots, x_n]$.
- Let N be an R-module.
- Let $G := \{f_1, \ldots, f_r\} \subset N$ be a finite subset of N.

Syzygies

Setup:

- Let K a field.
- Set $R := K[x_1, \ldots, x_n]$.
- Let N be an R-module.
- Let $G := \{f_1, \ldots, f_r\} \subset N$ be a finite subset of N.
- Let $F := R^r$ be the free R-module of rank r,
  and let $\mathrm{e}_1, \ldots, \mathrm{e}_r$ be the canonical basis of F.
- Let $\psi_G : F \to N$ be the homomorphism given by $\psi_G(\mathrm{e}_i) := f_i$.

Syzygies

Setup:

- Let K a field.
- Set $R := K[x_1, \dots, x_n]$.
- Let N be an R-module.
- Let $G := \{f_1, \dots, f_r\} \subset N$ be a finite subset of N.
- Let $F := R^r$ be the free R-module of rank r, and let $\mathrm{e}_1, \dots, \mathrm{e}_r$ be the canonical basis of F.
- Let $\psi_G : F \to N$ be the homomorphism given by $\psi_G(\mathrm{e}_i) := f_i$.

Definition

A **syzygy** of $G = \{f_1, \dots, f_r\}$ is an element of $\ker \psi_G$. We call

$$\mathrm{Syz}(G) := \ker \psi_G$$

the (first) **syzygy module** of G.

Syzygies

Example (1)

Consider $G := \{x, y, z\} \subset \mathbb{Q}[x, y, z]$. Then we have

$$\mathrm{Syz}(G) = \langle y\mathrm{e}_1 - x\mathrm{e}_2, z\mathrm{e}_1 - x\mathrm{e}_3, z\mathrm{e}_2 - y\mathrm{e}_3 \rangle = \left\langle \left(\begin{smallmatrix} y \\ -x \\ 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} z \\ 0 \\ -x \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 \\ z \\ -y \end{smallmatrix}\right) \right\rangle$$

and

$$\begin{pmatrix} x & y & z \end{pmatrix} \cdot \begin{pmatrix} y & z & 0 \\ -x & 0 & z \\ 0 & -x & -y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}.$$

Syzygies

Example (1)

Consider $G := \{x, y, z\} \subset \mathbb{Q}[x, y, z]$. Then we have

$$\mathrm{Syz}(G) = \langle y\mathrm{e}_1 - x\mathrm{e}_2, z\mathrm{e}_1 - x\mathrm{e}_3, z\mathrm{e}_2 - y\mathrm{e}_3 \rangle = \left\langle \left(\begin{smallmatrix} y \\ -x \\ 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} z \\ 0 \\ -x \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 \\ z \\ -y \end{smallmatrix}\right) \right\rangle$$

and

$$\begin{pmatrix} x & y & z \end{pmatrix} \cdot \begin{pmatrix} y & z & 0 \\ -x & 0 & z \\ 0 & -x & -y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}.$$

For computing syzygies, we use the following notation:

$$\begin{array}{c|ccc} x & y & z & 0 \\ y & -x & 0 & z \\ z & 0 & -x & -y \end{array}$$

Syzygies

Example (2)

$R = \mathbb{Q}[w, x, y, z]$

$$
\begin{array}{c|cccccc}
w^2 - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
wx - yz & w & -x & -y & 0 & z & z^2 \\
x^2 - wy & -z & w & 0 & -y & 0 & 0 \\
xy - z^2 & 0 & 0 & w & x & -y & -yz \\
y^2 - wz & 0 & 0 & -z & -w & x & w^2
\end{array}
$$

Free resolutions

Consider

- $G_1 := G$,
- G_{i+1}, $i > 1$, a finite set of generators of $\mathrm{Syz}(G_i)$, and
- $\phi_i := \psi_{G_i}$.

Inductively, we get a **free resolution**

$$\mathcal{F}:\ 0 \longleftarrow N/\langle G\rangle_R \longleftarrow N = F_0 \xleftarrow{\phi_1} F_1 \xleftarrow{\phi_2} F_2 \longleftarrow \dots$$
$$\dots \longleftarrow F_i \xleftarrow{\phi_{i+1}} F_{i+1} \longleftarrow \dots$$

Free resolutions

Consider

- $G_1 := G$,
- G_{i+1}, $i > 1$, a finite set of generators of $\mathrm{Syz}(G_i)$, and
- $\phi_i := \psi_{G_i}$.

Inductively, we get a **free resolution**

$$\mathcal{F}:\ 0 \longleftarrow N/\langle G\rangle_R \longleftarrow N = F_0 \xleftarrow{\phi_1} F_1 \xleftarrow{\phi_2} F_2 \longleftarrow \dots$$
$$\dots \longleftarrow F_i \xleftarrow{\phi_{i+1}} F_{i+1} \longleftarrow \dots$$

with

$$\operatorname{im}\phi_{i+1} = \operatorname{im}\psi_{G_{i+1}} = \langle G_{i+1}\rangle = \mathrm{Syz}(G_i) = \ker\psi_{G_i} = \ker\phi_i\,.$$

Free resolutions

Example

$R = \mathbb{Q}[w, x, y, z]$

$$
\begin{array}{c|cccccc}
w^2 - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
wx - yz & w & -x & -y & 0 & z & z^2 \\
x^2 - wy & -z & w & 0 & -y & 0 & 0 \\
xy - z^2 & 0 & 0 & w & x & -y & -yz \\
y^2 - wz & 0 & 0 & -z & -w & x & w^2 \\
\hline
 & 0 & y & -x & w & -z & 1 \\
 & -y^2 + wz & z^2 & -wy & yz & -w^2 & x
\end{array}
$$

Free resolutions

Example

$R = \mathbb{Q}[w, x, y, z]$

$$
\begin{array}{c|cccccc}
w^2 - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
wx - yz & w & -x & -y & 0 & z & z^2 \\
x^2 - wy & -z & w & 0 & -y & 0 & 0 \\
xy - z^2 & 0 & 0 & w & x & -y & -yz \\
y^2 - wz & 0 & 0 & -z & -w & x & w^2 \\
\hline
 & 0 & y & -x & w & -z & 1 \\
 & -y^2 + wz & z^2 & -wy & yz & -w^2 & x
\end{array}
$$

We thus get

$$
0 \leftarrow R/\langle G \rangle_R \leftarrow R \xleftarrow{\phi_1} R(-2)^5 \xleftarrow{\phi_2} \begin{array}{c} R(-3)^5 \\ \oplus \\ R(-4) \end{array} \xleftarrow{\phi_3} \begin{array}{c} R(-4) \\ \oplus \\ R(-5) \end{array} \leftarrow 0\,.
$$

Betti tables

Example

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

Betti tables

Example

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

$$0 \leftarrow R/I \leftarrow R \leftarrow \begin{matrix} R(-2)^{18} \\ \oplus \\ R(-3)^{13} \end{matrix} \leftarrow \begin{matrix} R(-3)^{55} \\ \oplus \\ R(-4)^{75} \end{matrix} \leftarrow \begin{matrix} R(-4)^{75} \\ \oplus \\ R(-5)^{180} \end{matrix} \leftarrow \dots$$

Betti tables

Example

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

$$0 \leftarrow R/I \leftarrow R \leftarrow \begin{array}{c} R(-2)^{18} \\ \oplus \\ R(-3)^{13} \end{array} \leftarrow \begin{array}{c} R(-3)^{55} \\ \oplus \\ R(-4)^{75} \end{array} \leftarrow \begin{array}{c} R(-4)^{75} \\ \oplus \\ R(-5)^{180} \end{array} \leftarrow \dots$$

Prym-canonical nodal curve of genus 10,
given by a homogeneous ideal $I \subseteq R := \mathbb{F}_p[x_0, \dots, x_8]$.

Outline

Induced ordering

Definition

Let $>$ be a monomial ordering on $F_0 = R^s$ and consider a set $G := \{f_1, \ldots, f_r\} \subset F_0 \setminus \{0\}$.
Then the **induced ordering** $\succ$ on $F_1 := R^r$ (w.r.t. $>$ and G) is given by

$$m_1 \mathrm{e}_i \succ m_2 \mathrm{e}_j \;:\Leftrightarrow\; \mathsf{LT}(m_1 f_i) > \mathsf{LT}(m_2 f_j)$$
$$\text{or } (\mathsf{LT}(m_1 f_i) = \mathsf{LT}(m_2 f_j) \text{ and } i > j)\,.$$

Induced ordering

Example

$R = \mathbb{Q}[w, x, y, z]$ with the degree reverse lexicographical ordering

$$\begin{array}{r|cccccc}
\mathbf{w^2} - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
\mathbf{wx} - yz & w & -x & -y & 0 & z & z^2 \\
\mathbf{x^2} - wy & -z & w & 0 & -y & 0 & 0 \\
\mathbf{xy} - z^2 & 0 & 0 & w & x & -y & -yz \\
\mathbf{y^2} - wz & 0 & 0 & -z & -w & x & w^2 \\
\hline
 & 0 & y & -x & w & -z & 1 \\
 & -y^2 + wz & z^2 & -wy & yz & -w^2 & x
\end{array}$$

Induced ordering

Example

$R = \mathbb{Q}[w, x, y, z]$ with the degree reverse lexicographical ordering

$$
\begin{array}{r|cccccc}
\boldsymbol{w^2} - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
\boldsymbol{wx} - yz & w & -x & -y & 0 & z & z^2 \\
\boldsymbol{x^2} - wy & -z & w & 0 & -y & 0 & 0 \\
\boldsymbol{xy} - z^2 & 0 & 0 & w & x & -y & -yz \\
\boldsymbol{y^2} - wz & 0 & 0 & -z & -w & x & w^2 \\
\hline
 & 0 & y & -x & w & -z & 1 \\
 & -y^2 + wz & z^2 & -wy & yz & -w^2 & x
\end{array}
$$

Induced ordering

Example

$R = \mathbb{Q}[w, x, y, z]$ with the degree reverse lexicographical ordering

$$
\begin{array}{r|cccccc}
\boldsymbol{w^2} - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
\boldsymbol{wx} - yz & \boldsymbol{w} & -x & -y & 0 & z & z^2 \\
\boldsymbol{x^2} - wy & -z & \boldsymbol{w} & 0 & -y & 0 & 0 \\
\boldsymbol{xy} - z^2 & 0 & 0 & \boldsymbol{w} & \boldsymbol{x} & -y & -yz \\
\boldsymbol{y^2} - wz & 0 & 0 & -z & -w & \boldsymbol{x} & \boldsymbol{w^2} \\
\hline
 & 0 & y & -x & w & -z & 1 \\
 & -y^2 + wz & z^2 & -wy & yz & -w^2 & x
\end{array}
$$

Induced ordering

Example

$R = \mathbb{Q}[w, x, y, z]$ with the degree reverse lexicographical ordering

$$
\begin{array}{c|cccccc}
\mathbf{w^2} - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
\mathbf{wx} - yz & \mathbf{w} & -x & -y & 0 & z & z^2 \\
\mathbf{x^2} - wy & -z & \mathbf{w} & 0 & -y & 0 & 0 \\
\mathbf{xy} - z^2 & 0 & 0 & \mathbf{w} & \mathbf{x} & -y & -yz \\
\mathbf{y^2} - wz & 0 & 0 & -z & -w & \mathbf{x} & \mathbf{w^2} \\
\hline
 & 0 & y & -x & w & -z & 1 \\
 & -y^2 + wz & z^2 & -wy & yz & -w^2 & x
\end{array}
$$

Induced ordering

Example

$R = \mathbb{Q}[w, x, y, z]$ with the degree reverse lexicographical ordering

$$
\begin{array}{r|cccccc}
\boldsymbol{w^2} - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
\boldsymbol{wx} - yz & \boldsymbol{w} & -x & -y & 0 & z & z^2 \\
\boldsymbol{x^2} - wy & -z & \boldsymbol{w} & 0 & -y & 0 & 0 \\
\boldsymbol{xy} - z^2 & 0 & 0 & \boldsymbol{w} & \boldsymbol{x} & -y & -yz \\
\boldsymbol{y^2} - wz & 0 & 0 & -z & -w & \boldsymbol{x} & \boldsymbol{w^2} \\
\hline
 & 0 & y & -x & \boldsymbol{w} & -z & 1 \\
 & -y^2 + wz & z^2 & -wy & yz & -w^2 & \boldsymbol{x}
\end{array}
$$

S-pairs und syzygies

Every zero-reduction of an s-pair in Buchberger's algorithm yields a syzygy:

S-pairs und syzygies

Every zero-reduction of an s-pair in Buchberger's algorithm yields a syzygy:

Let

$$S(f_i, f_j) := m_{ji} f_i - m_{ij} f_j = g_1^{(ij)} f_1 + \ldots + g_r^{(ij)} f_r$$

with

$$m_{ji} := \frac{\mathrm{LCM}(\mathrm{LM}(f_j), \mathrm{LM}(f_i))}{\mathrm{LT}(f_i)} \in R$$

be a *standard representation* of $S(f_i, f_j)$.

S-pairs und syzygies

Every zero-reduction of an s-pair in Buchberger's algorithm yields a syzygy:

Let

$$S(f_i, f_j) := m_{ji} f_i - m_{ij} f_j = g_1^{(ij)} f_1 + \ldots + g_r^{(ij)} f_r$$

with

$$m_{ji} := \frac{\mathsf{LCM}(\mathsf{LM}(f_j), \mathsf{LM}(f_i))}{\mathsf{LT}(f_i)} \in R$$

be a *standard representation* of $S(f_i, f_j)$.

Then

$$m_{ji} \mathrm{e}_i - m_{ij} \mathrm{e}_j - \left(g_1^{(ij)} \mathrm{e}_1 + \ldots + g_r^{(ij)} \mathrm{e}_r \right)$$

is an element of $\mathsf{Syz}(G)$ for $G := \{f_1, \ldots, f_r\} \subset F_0 \setminus \{0\}$ as above.

S-pairs und syzygies

Every zero-reduction of an s-pair in Buchberger's algorithm yields a syzygy:

$$S(f_i, f_j) := m_{ji} f_i - m_{ij} f_j = g_1^{(ij)} f_1 + \ldots + g_r^{(ij)} f_r$$

$$\downarrow$$

$$m_{ji}\mathrm{e}_i - m_{ij}\mathrm{e}_j - \left(g_1^{(ij)}\mathrm{e}_1 + \ldots + g_r^{(ij)}\mathrm{e}_r\right)$$

$$\in \mathsf{Syz}(G)$$

Schreyer's theorem

Theorem

Let $G = \{f_1, \ldots, f_r\} \subset F_0 := R^s$ *be a Gröbner basis w.r.t. a monomial ordering* $>$ *on* F_0*. For every pair* (f_i, f_j) *with* $i, j \in \{1, \ldots, r\}$ *let*

$$S(f_i, f_j) = m_{ji} f_i - m_{ij} f_j = g_1^{(ij)} f_1 + \ldots + g_r^{(ij)} f_r$$

be a standard representation of the corresponding s-pair.

Schreyer's theorem

Theorem

Let $G = \{f_1, \ldots, f_r\} \subset F_0 := R^s$ be a Gröbner basis w.r.t. a monomial ordering $>$ on F_0. For every pair (f_i, f_j) with $i, j \in \{1, \ldots, r\}$ let

$$S(f_i, f_j) = m_{ji} f_i - m_{ij} f_j = g_1^{(ij)} f_1 + \ldots + g_r^{(ij)} f_r$$

be a standard representation of the corresponding s-pair. Then the relations

$$m_{ji}\mathrm{e}_i - m_{ij}\mathrm{e}_j - \left(g_1^{(ij)}\mathrm{e}_1 + \ldots + g_r^{(ij)}\mathrm{e}_r\right) \in F_1 := R^r$$

form a Gröbner basis of $\mathrm{Syz}(G)$ *w.r.t. the monomial ordering on* F_1 *induced by* $>$ *and* G.
In particular, they generate the syzygy module $\mathrm{Syz}(G)$.

Schreyer's theorem

Example

$$\begin{array}{r|cccccc}
\mathbf{w^2} - xz & -x & y & 0 & -z & 0 & -y^2 + wz \\
\mathbf{wx} - yz & \mathbf{w} & -x & -y & 0 & z & z^2 \\
\mathbf{x^2} - wy & -z & \mathbf{w} & 0 & -y & 0 & 0 \\
\mathbf{xy} - z^2 & 0 & 0 & \mathbf{w} & \mathbf{x} & -y & -yz \\
\mathbf{y^2} - wz & 0 & 0 & -z & -w & \mathbf{x} & \mathbf{w^2}
\end{array}$$

Schreyer's algorithm

Input: a Gröbner basis $G = \{f_1, \dots, f_r\} \subset F_0 := R^s$
w.r.t. a monomial ordering $>$
Output: a Gröbner basis of $\mathrm{Syz}(G) \subset F_1 := R^r$
w.r.t. the monomial ordering induced by $>$ and G

Schreyer's algorithm

Input: a Gröbner basis $G = \{f_1, \dots, f_r\} \subset F_0 := R^s$
w.r.t. a monomial ordering $>$
Output: a Gröbner basis of $\mathrm{Syz}(G) \subset F_1 := R^r$
w.r.t. the monomial ordering induced by $>$ and G

for each pair (i, j) with $1 \leq j < i \leq r$ **do**
    $h := S(f_i, f_j) = m_{ji} f_i - m_{ij} f_j \in F_0$
    while $h \neq 0$ **do**
        choose an index λ such that $\mathrm{LT}(f_\lambda)$ divides $\mathrm{LT}(h)$
        $h := h - \frac{\mathrm{LT}(h)}{\mathrm{LT}(f_\lambda)} f_\lambda$

Schreyer's algorithm

Input: a Gröbner basis $G = \{f_1, \dots, f_r\} \subset F_0 := R^s$
w.r.t. a monomial ordering $>$

Output: a Gröbner basis of $\mathrm{Syz}(G) \subset F_1 := R^r$
w.r.t. the monomial ordering induced by $>$ and G

```
S := ∅
for each pair (i, j) with 1 ≤ j < i ≤ r do
    h := S(f_i, f_j) = m_ji f_i − m_ij f_j ∈ F_0
    s := m_ji e_i − m_ij e_j ∈ F_1
    while h ≠ 0 do
        choose an index λ such that LT(f_λ) divides LT(h)
        h := h − (LT(h) / LT(f_λ)) f_λ
        s := s − (LT(h) / LT(f_λ)) e_λ
    S := S ∪ {s}
return S
```

Schreyer frame

- For a syzygy of the form

$$m_{ji}\mathrm{e}_i - m_{ij}\mathrm{e}_j - \left(g_1^{(ij)}\mathrm{e}_1 + \ldots + g_r^{(ij)}\mathrm{e}_r\right)$$

with $i > j$, we have, w.r.t. the induced monomial ordering,

$$m_{ji}\mathrm{e}_i \succ m_{ij}\mathrm{e}_j \succ \mathsf{LM}\left(g_k^{(ij)}\right)\mathrm{e}_k$$

for all $k = 1, \ldots, r$ with $g_k^{(ij)} \neq 0$.

Schreyer frame

- For a syzygy of the form

$$m_{ji}\mathrm{e}_i - m_{ij}\mathrm{e}_j - \left(g_1^{(ij)}\mathrm{e}_1 + \ldots + g_r^{(ij)}\mathrm{e}_r\right)$$

  with $i > j$, we have, w.r.t. the induced monomial ordering,

$$m_{ji}\mathrm{e}_i \succ m_{ij}\mathrm{e}_j \succ \mathrm{LM}\left(g_k^{(ij)}\right)\mathrm{e}_k$$

  for all $k = 1, \ldots, r$ with $g_k^{(ij)} \neq 0$.

- Therefore the leading terms of $\mathrm{Syz}(G)$ only depend on the leading terms $\mathrm{LT}(G)$ of G, and not on the remaining terms.

Schreyer frame

- For a syzygy of the form

$$m_{ji}\mathrm{e}_i - m_{ij}\mathrm{e}_j - \left(g_1^{(ij)}\mathrm{e}_1 + \ldots + g_r^{(ij)}\mathrm{e}_r\right)$$

with $i > j$, we have, w.r.t. the induced monomial ordering,

$$m_{ji}\mathrm{e}_i \succ m_{ij}\mathrm{e}_j \succ \mathsf{LM}\left(g_k^{(ij)}\right)\mathrm{e}_k$$

for all $k = 1, \ldots, r$ with $g_k^{(ij)} \neq 0$.

- Therefore the leading terms of Syz(G) only depend on the leading terms LT(G) of G, and not on the remaining terms.

- Starting from LT(G), we may thus inductively compute the leading terms of all syzygies in the free resolution of G.

Schreyer frame

Example

w^2						
wx	w					
x^2		w				
xy			w	x		
y^2					x	w^2
				w		
						x

Outline

Computing syzygies
- Syzygies and free resolutions
- Schreyer's algorithm
- Improvements

Application: Prym-Green Conjecture
- The Prym-Green Conjecture
- Computing Prym-Green matrices
- The structure of Prym-Green matrices

Lower order terms

First observation:
Those terms which are not in the leading ideal $\mathsf{L}(G) = \langle \mathsf{LT}(G) \rangle$ do not contribute to the result and can be left out.

Lower order terms

First observation:
Those terms which are not in the leading ideal $\mathsf{L}(G) = \langle \mathsf{LT}(G) \rangle$ do not contribute to the result and can be left out.

Definition

Let S be a subset of R^s and let $t \in R^s$ be a term. Then t is called a **lower order term** w.r.t. S if

$$\mathsf{LM}(f) \nmid t \quad \text{for all } f \in S \setminus \{0\}\,.$$

Ordering terms

Second observation:
If lower order terms are left out, then the remaining terms in the intermediate steps do not have to be ordered.

Ordering terms

Second observation:
If lower order terms are left out, then the remaining terms in the intermediate steps do not have to be ordered.

- The whole computation may thus be performed without monomial comparisons.

Ordering terms

Second observation:
If lower order terms are left out, then the remaining terms in the intermediate steps do not have to be ordered.

- The whole computation may thus be performed without monomial comparisons.
- Monomial comparisons w.r.t. the induced ordering are expensive, in particular towards the end of the resolution.

Ordering terms

Second observation:
If lower order terms are left out, then the remaining terms in the intermediate steps do not have to be ordered.

- The whole computation may thus be performed without monomial comparisons.
- Monomial comparisons w.r.t. the induced ordering are expensive, in particular towards the end of the resolution.
- If the terms are not ordered in the intermediate steps, then the algorithm resembles a tree structure:

Tree structure

Example

$R = \mathbb{Q}[w, x, y, z]$ with the lexicographical ordering

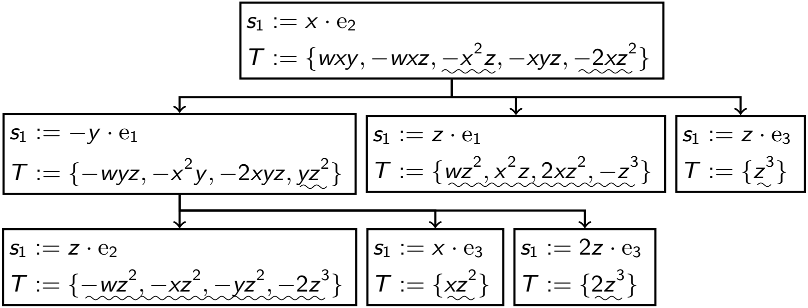
$$\left.\begin{array}{l} \boldsymbol{wx} + wz + x^2 + 2xz - z^2 \\ \boldsymbol{wy} - wz - xz - yz - 2z^2 \\ \boldsymbol{xy} + z^2 \end{array}\right| \begin{array}{ll} \boldsymbol{x} & \\ & \boldsymbol{w} \end{array}$$

Tree structure

Example

$R = \mathbb{Q}[w, x, y, z]$ with the lexicographical ordering

$$\begin{array}{l|ll}
\boldsymbol{wx} + wz + x^2 + 2xz - z^2 & & \\
\boldsymbol{wy} - wz - xz - yz - 2z^2 & \boldsymbol{x} & \\
\boldsymbol{xy} + z^2 & & \boldsymbol{w}
\end{array}$$



Tree structure

Example (continued)

$$\bar{s}_2 := w \cdot \mathrm{e}_3$$
$$T := \{wxy, \utilde{wz}^2\}$$

$$\downarrow$$

cached:

$$\bar{s}_2 := -y \cdot \mathrm{e}_1 + z \cdot \mathrm{e}_2 + x \cdot \mathrm{e}_3 + 2z \cdot \mathrm{e}_3$$
$$T := \varnothing$$

Tree structure

Example (continued)

$\bar{s}_2 := w \cdot \mathrm{e}_3$

$T := \{wxy, \underset{\sim}{wz^2}\}$

$\downarrow$

cached:

$\bar{s}_2 := -y \cdot \mathrm{e}_1 + z \cdot \mathrm{e}_2 + x \cdot \mathrm{e}_3 + 2z \cdot \mathrm{e}_3$

$T := \varnothing$

$$\begin{array}{l|cc} \boldsymbol{wx} + wz + x^2 + 2xz - z^2 & -y + z & -y \\ \boldsymbol{wy} - wz - xz - yz - 2z^2 & \boldsymbol{x} + z & z \\ \boldsymbol{xy} + z^2 & x + 3z & \boldsymbol{w} + x + 2z \end{array}$$

Tree structure

Example (continued)

$\bar{s}_2 := w \cdot \mathrm{e}_3$

$T := \{wxy, \underset{\sim}{wz^2}\}$

$\downarrow$

cached:

$\bar{s}_2 := -y \cdot \mathrm{e}_1 + z \cdot \mathrm{e}_2 + x \cdot \mathrm{e}_3 + 2z \cdot \mathrm{e}_3$

$T := \varnothing$

$$
\begin{array}{l|ll}
\boldsymbol{wx} + wz + x^2 + 2xz - z^2 & -y + z & -y \\
\boldsymbol{wy} - wz - xz - yz - 2z^2 & \boldsymbol{x} + z & z \\
\boldsymbol{xy} + z^2 & x + 3z & \boldsymbol{w} + x + 2z
\end{array}
$$

Third observation:
Because of the tree structure, intermediate results can be cached and reused.

Summary

Improvements for Schreyer's algorithm:

1. Lower order terms can be left out.

Summary

Improvements for Schreyer's algorithm:

1. Lower order terms can be left out.
2. The terms do not have to be ordered.

Summary

Improvements for Schreyer's algorithm:

1. Lower order terms can be left out.
2. The terms do not have to be ordered.
3. Intermediate results can be cached and reused.

Timings

	Macaulay2	Singular	
g	`res`	`lres`	`s_res`
7	0.02	0.01	0.00
8	0.18	0.05	0.01
9	2.96	0.75	0.04
10	64.78	19.56	0.15
11	901.14	344.62	0.64
12	16,597.90	6,261.13	3.43
13	–	–	25.95
14	–	–	224.15
15	–	–	2,002.51
16	–	–	18,612.82

PCNC, in seconds

Eröcal, Motsak, Schreyer, S.:
Refined Algorithms to Compute Syzygies (JSC 2016)

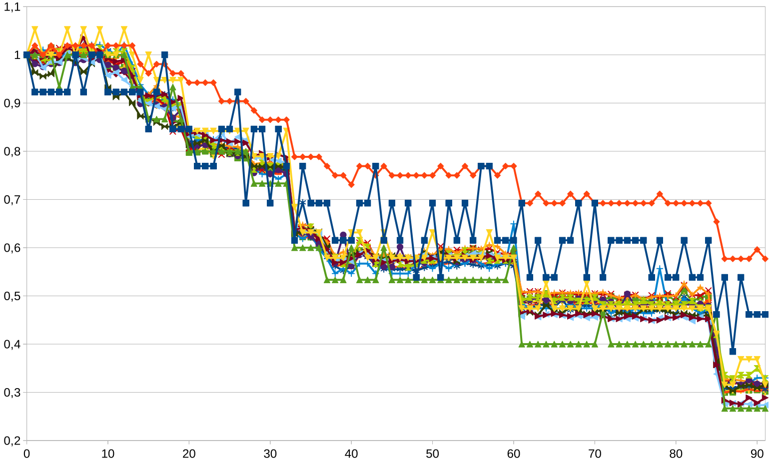
Timings

	Macaulay2	Singular		
g	`res`	`lres`	`s_res`	`fres`
7	0.02	0.01	0.00	0.00
8	0.18	0.05	0.01	0.01
9	2.96	0.75	0.04	0.02
10	64.78	19.56	0.15	0.05
11	901.14	344.62	0.64	0.16
12	16,597.90	6,261.13	3.43	0.51
13	–	–	25.95	1.56
14	–	–	224.15	5.25
15	–	–	2,002.51	15.90
16	–	–	18,612.82	48.89

PCNC, in seconds

Eröcal, Motsak, Schreyer, S.:
Refined Algorithms to Compute Syzygies (JSC 2016)

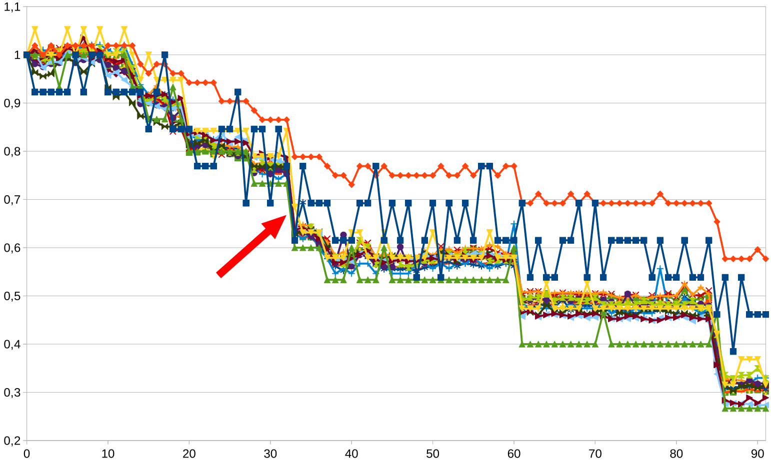
Restructuring the code



by changing a single line

18,612 s ⟶ 160 s

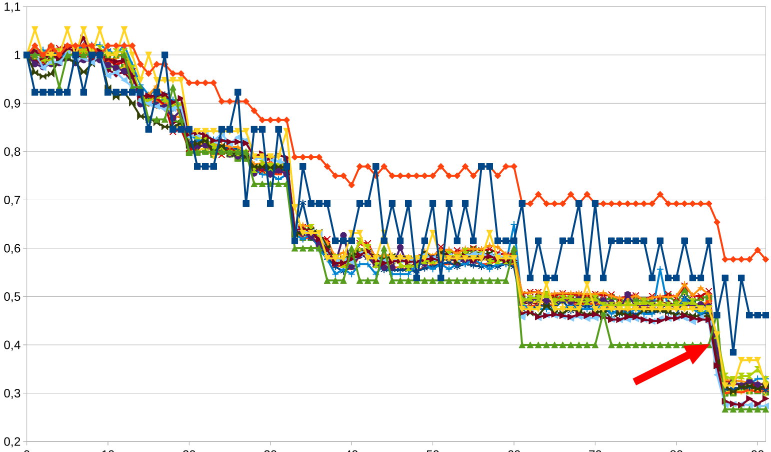
Restructuring the code



search for reducers in the right order

improvement: 0.134

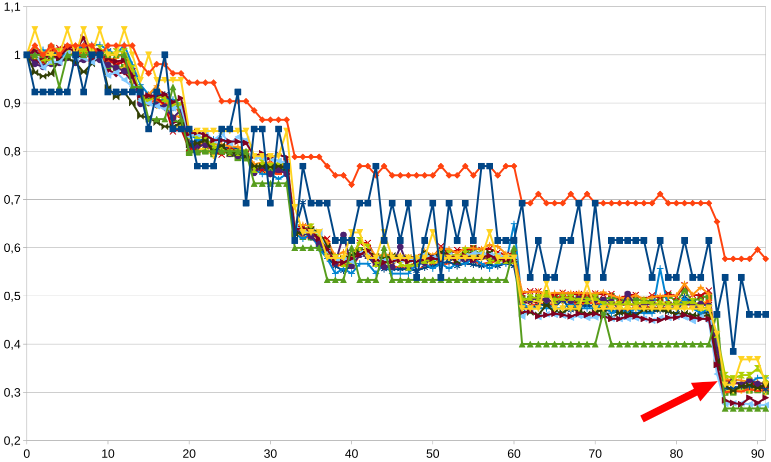
Restructuring the code



use C array instead of `std::map` from C++ to hash reducers

improvement: 0.120

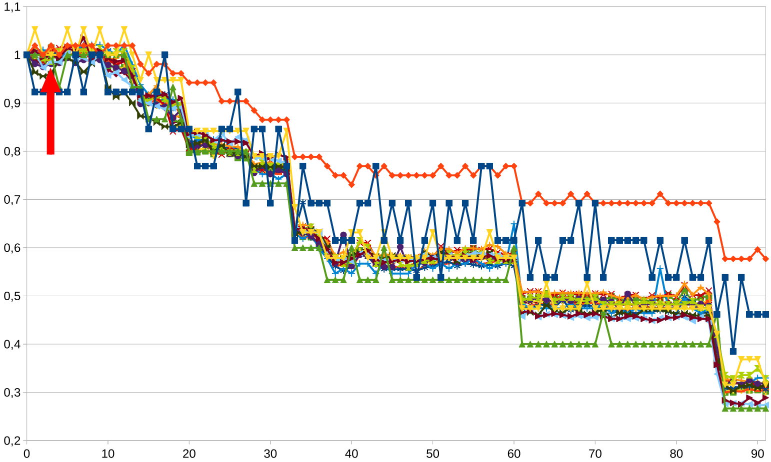
Restructuring the code



use C array instead of `std::vector` from C++ to hash reducers

improvement: 0.058

Restructuring the code



remove some global variable

regression: 0.02

Outline

Prym-canonical nodal curves

Example

Let C be a Prym-canonical nodal curve of genus 10, given by a homogeneous ideal $I \subseteq R := \mathbb{F}_p[x_0, \ldots, x_8]$.

Prym-canonical nodal curves

Example

Let C be a Prym-canonical nodal curve of genus 10, given by a homogeneous ideal $I \subseteq R := \mathbb{F}_p[x_0, \ldots, x_8]$.

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

Prym-canonical nodal curves

Example

Let C be a Prym-canonical nodal curve of genus 10, given by a homogeneous ideal $I \subseteq R := \mathbb{F}_p[x_0, \ldots, x_8]$.

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

$$0 \leftarrow R/I \leftarrow R \leftarrow \begin{matrix} R(-2)^{18} \\ \oplus \\ R(-3)^{13} \end{matrix} \leftarrow \begin{matrix} R(-3)^{55} \\ \oplus \\ R(-4)^{75} \end{matrix} \leftarrow \begin{matrix} R(-4)^{75} \\ \oplus \\ R(-5)^{180} \end{matrix} \leftarrow \ldots$$

Prym-canonical nodal curves

Example

Let C be a Prym-canonical nodal curve of genus 10, given by a homogeneous ideal $I \subseteq R := \mathbb{F}_p[x_0, \ldots, x_8]$.

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

$$0 \leftarrow R/I \leftarrow R \leftarrow \begin{array}{c} R(-2)^{18} \\ \oplus \\ R(-3)^{13} \end{array} \leftarrow \begin{array}{c} R(-3)^{55} \\ \oplus \\ R(-4)^{75} \end{array} \swarrow \begin{array}{c} R(-4)^{75} \\ \oplus \\ R(-5)^{180} \end{array} \leftarrow \ldots$$

Prym-canonical nodal curves

Example

Let C be a Prym-canonical nodal curve of genus 10, given by a homogeneous ideal $I \subseteq R := \mathbb{F}_p[x_0, \ldots, x_8]$.

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

$$0 \leftarrow R/I \leftarrow R \leftarrow \begin{matrix} R(-2)^{18} \\ \oplus \\ R(-3)^{13} \end{matrix} \leftarrow \begin{matrix} R(-3)^{55} \\ \oplus \\ R(-4)^{75} \end{matrix} \swarrow \begin{matrix} R(-4)^{75} \\ \oplus \\ R(-5)^{180} \end{matrix} \leftarrow \ldots$$

C is Gorenstein.

Prym-canonical nodal curves

Example

Let C be a Prym-canonical nodal curve of genus 10, given by a homogeneous ideal $I \subseteq R := \mathbb{F}_p[x_0, \ldots, x_8]$.

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	42	-	-	-	-	-
2:	-	-	-	126	210	162	63	10
Σ:	1	18	42	126	210	162	63	10

C admits a **pure resolution**.

Prym-Green conjecture

Prym-canonical nodal curves of even genus admit a pure resolution.

Prym-Green conjecture

Prym-canonical nodal curves of even genus admit a pure resolution.

exceptions: $g = 8$,

	0	1	2	3	4	5
0:	1	-	-	-	-	-
1:	-	7	1	-	-	-
2:	-	1	35	56	35	8
Σ:	1	8	36	56	35	8

Prym-Green conjecture

Prym-canonical nodal curves of even genus admit a pure resolution.

exceptions: $g = 8, 16,$

	0	1	2	3	4	5	6	7	8	...
0:	1	-	-	-	-	-	-	-	-	
1:	-	75	520	1755	3432	3575	1	-	-	
2:	-	-	-	-	-	1	6435	11440	11583	...
Σ:	1	75	520	1755	3432	3576	6436	11440	11583	...

Prym-Green conjecture

Prym-canonical nodal curves of even genus admit a pure resolution.

exceptions: $g = 8, 16, ??$

Prym-Green conjecture

Prym-canonical nodal curves of even genus admit a pure resolution.

exceptions: $g = 8, 16, ??$

Algorithmic solution:

- compute resolution
- reduce matrix

Prym-Green conjecture

Prym-canonical nodal curves of even genus admit a pure resolution.

exceptions: $g = 8, 16, ??$

Algorithmic solution:

- compute resolution
- reduce matrix

For $g = 24$, the Prym-Green matrix has size

$$3.250.026 \times 3.250.026\,!$$

Prym-Green conjecture

Prym-canonical nodal curves of even genus admit a pure resolution.

exceptions: $g = 8, 16, ??$

Algorithmic solution:

- compute resolution
- reduce matrix

For $g = 24$, the Prym-Green matrix has size

$$3.250.026 \times 3.250.026\,!$$

Outline

Computing Prym-Green matrices

$$I = \langle \underbrace{g_1, \ldots, g_s}_{\text{GB}} \rangle \subset R := K[x_1, \ldots, x_n] \quad \text{and}$$

$$x_n \nmid \mathrm{LT}(g_j) \quad \forall j = 1, \ldots, s$$

Computing Prym-Green matrices

$$I = \langle \underbrace{g_1, \dots, g_s}_{\text{GB}} \rangle \subset R := K[x_1, \dots, x_n] \quad \text{and}$$

$$x_n \nmid \mathrm{LT}(g_j) \quad \forall j = 1, \dots, s$$

$\Rightarrow x_n$ is a non-zerodivisor in R/I.

Computing Prym-Green matrices

$$I = \langle \underbrace{g_1, \ldots, g_s}_{\text{GB}} \rangle \subset R := K[x_1, \ldots, x_n] \quad \text{and}$$

$$x_n \nmid \mathrm{LT}(g_j) \quad \forall j = 1, \ldots, s$$

$\Rightarrow x_n$ is a non-zerodivisor in R/I.

$\Rightarrow (0 \longleftarrow R/I \longleftarrow R \longleftarrow R^{r_1} \longleftarrow \ldots) \otimes_R R/\langle x_n \rangle$ is exact.

Computing Prym-Green matrices

$$I = \langle \underbrace{g_1, \ldots, g_s}_{\text{GB}} \rangle \subset R := K[x_1, \ldots, x_n] \quad \text{and}$$

$$x_n \nmid \mathrm{LT}(g_j) \quad \forall j = 1, \ldots, s$$

$\Rightarrow x_n$ is a non-zerodivisor in R/I.

$$\Rightarrow (0 \longleftarrow R/I \longleftarrow R \longleftarrow R^{r_1} \longleftarrow \ldots) \otimes_R R/\langle x_n \rangle \text{ is exact.}$$

improvement for $g = 16$: 48.9 s $\longrightarrow$ 9.9 s

Computing Prym-Green matrices

$$I = \langle \underbrace{g_1, \dots, g_s}_{\text{GB}} \rangle \subset R := K[x_1, \dots, x_n] \quad \text{and}$$

$$x_n \nmid \mathrm{LT}(g_j) \quad \forall j = 1, \dots, s$$

$\Rightarrow x_n$ is a non-zerodivisor in R/I.

$\Rightarrow (0 \longleftarrow R/I \longleftarrow R \longleftarrow R^{r_1} \longleftarrow \dots) \otimes_R R/\langle x_n \rangle$ is exact.

improvement for $g = 16$: $48.9\text{ s} \longrightarrow 9.9\text{ s}$

with some more tricks: 1.8 s

Computing Prym-Green matrices

$$I = \langle \underbrace{g_1, \dots, g_s}_{\text{GB}} \rangle \subset R := K[x_1, \dots, x_n] \quad \text{and}$$

$$x_n \nmid \mathrm{LT}(g_j) \quad \forall j = 1, \dots, s$$

$\Rightarrow x_n$ is a non-zerodivisor in R/I.

$\Rightarrow (0 \longleftarrow R/I \longleftarrow R \longleftarrow R^{r_1} \longleftarrow \dots) \otimes_R R/\langle x_n \rangle$ is exact.

improvement for $g = 16$: 48.9 s $\longrightarrow 9.9$ s

with some more tricks: 1.8 s

as a reminder:
Eröcal, Motsak, Schreyer, S.: Refined Algorithms to Compute Syzygies (JSC 2016): 18.612 s

Project status

Prym-Green matrix for $g = 24$

- Time to compute: 20.768 s

Project status

Prym-Green matrix for $g = 24$

- Time to compute: 20.768 s

Algorithmic solution:

- compute resolution ✔
- reduce matrix

Project status

Prym-Green matrix for $g = 24$

- Time to compute: 20.768 s
- Number of entries: 8.616.423.582

Algorithmic solution:

- compute resolution ✔
- reduce matrix

Project status

Prym-Green matrix for $g = 24$

- Time to compute: 20.768 s
- Number of entries: 8.616.423.582
- Amount of memory
  (in the SINGULAR kernel, 40 Byte/entry): 321 GB

Algorithmic solution:

- compute resolution ✔
- reduce matrix

Project status

Prym-Green matrix for $g = 24$

- Time to compute: 20.768 s
- Number of entries: 8.616.423.582
- Amount of memory (in the SINGULAR kernel, 40 Byte/entry): 321 GB

Algorithmic solution:

- compute resolution ✔
- save matrix
- reduce matrix

Outline

Computing syzygies

Syzygies and free resolutions

Schreyer's algorithm

Improvements

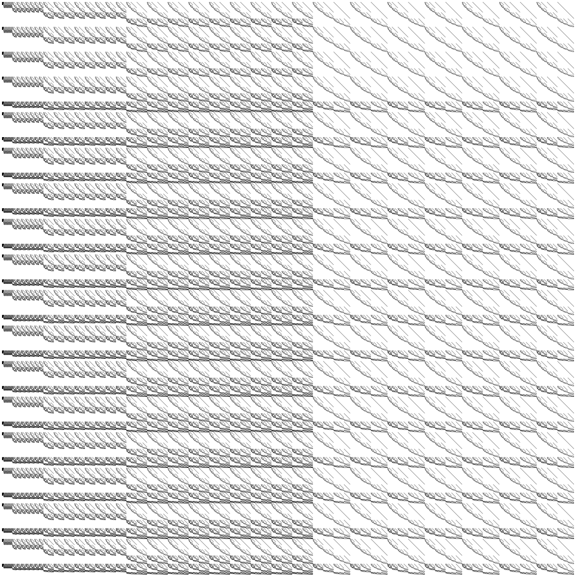
Application: Prym-Green Conjecture

The Prym-Green Conjecture

Computing Prym-Green matrices

The structure of Prym-Green matrices

Prym-Green matrix for $g = 14$ (1932×1932)



Koszul complex

Definition (Koszul complex, special case)

Given $R = K[x_1, \ldots, x_n]$ as before, we define the **Koszul complex** $\mathcal{K}_n = K(x_1, \ldots, x_n)$ to be the minimal free resolution of $R/\langle x_1, \ldots, x_n \rangle$.

Koszul complex

Definition (Koszul complex, special case)

Given $R = K[x_1, \ldots, x_n]$ as before, we define the **Koszul complex** $\mathcal{K}_n = K(x_1, \ldots, x_n)$ to be the minimal free resolution of $R/\langle x_1, \ldots, x_n \rangle$.

Note that $K(x_1, \ldots, x_n)$ is well-defined up to isomorphism of complexes.

Koszul complex

Definition (Koszul complex, special case)

Given $R = K[x_1, \ldots, x_n]$ as before, we define the **Koszul complex** $\mathcal{K}_n = K(x_1, \ldots, x_n)$ to be the minimal free resolution of $R/\langle x_1, \ldots, x_n \rangle$.

Note that $K(x_1, \ldots, x_n)$ is well-defined up to isomorphism of complexes.

Example

For $n = 3$, we have

$$\mathcal{K}_3 : 0 \longleftarrow R/\langle x_1, x_2, x_3 \rangle \longleftarrow R \xleftarrow{\phi_1} R^3 \xleftarrow{\phi_2} R^3 \xleftarrow{\phi_3} R \longleftarrow 0$$

with $\phi_1 = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}, \quad \phi_2 = \begin{pmatrix} -x_2 & -x_3 & 0 \\ x_1 & 0 & -x_3 \\ 0 & x_1 & x_2 \end{pmatrix}, \quad \phi_3 = \begin{pmatrix} x_3 \\ -x_2 \\ x_1 \end{pmatrix}.$

Koszul complex

Example

```
> ring R = 0, x(1..3), dp;
> ideal I = x(1..3);
> resolution S = fres(I, 0);
> print(betti(S, 0), "betti");
           0     1     2     3
------------------------------
    0:     1     3     3     1
------------------------------
total:     1     3     3     1

> print(S[2]);
-x(2),-x(3),0,
x(1), 0,    -x(3),
0,    x(1), x(2)
```

Koszul complex

Example

```
> ring R = 0, x(1..5), dp;
> ideal I = x(1..5);
> resolution S = fres(I, 0);
> S;
 1      5      10      10      5      1
R <--  R <--  R <--   R <--   R <--  R

0      1      2       3       4      5
resolution not minimized yet
> print(betti(S, 0), "betti");
           0     1     2     3     4     5
------------------------------------------
    0:     1     5    10    10     5     1
------------------------------------------
total:     1     5    10    10     5     1
```

Koszul complex

Example

```
> print(S[3]);
x(3), x(4), 0,    0,    x(5), 0,    0,    0,    0,    0,
-x(2),0,    x(4), 0,    0,    x(5), 0,    0,    0,    0,
x(1), 0,    0,    x(4), 0,    0,    x(5), 0,    0,    0,
0,    -x(2),-x(3),0,    0,    0,    0,    x(5), 0,    0,
0,    x(1), 0,    -x(3),0,    0,    0,    0,    x(5), 0,
0,    0,    x(1), x(2), 0,    0,    0,    0,    0,    x(5),
0,    0,    0,    0,    -x(2),-x(3),0,    -x(4),0,    0,
0,    0,    0,    0,    x(1), 0,    -x(3),0,    -x(4),0,
0,    0,    0,    0,    0,    x(1), x(2), 0,    0,    -x(4),
0,    0,    0,    0,    0,    0,    0,    x(1), x(2), x(3)
```

Betti tables of Prym-canonical nodal curves

Write the rows in the (non-minimal) Betti tables of Prym-canonical nodal curves as linear combinations of the rows in Pascal's triangle:

$g = 10$

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

1						
1	1					
1	2	1				
1	3	3	1			
1	4	6	4	1		
1	5	10	10	5	1	
1	6	15	20	15	6	1

Betti tables of Prym-canonical nodal curves

Write the rows in the (non-minimal) Betti tables of Prym-canonical nodal curves as linear combinations of the rows in Pascal's triangle:

$\boldsymbol{g = 10}$

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

=		1							
		1	1						
		1	2	1					
		1	3	3	1				
		1	4	6	4	1			
		1	5	10	10	5	1		·3
	+	1	6	15	20	15	6	1	·10

Betti tables of Prym-canonical nodal curves

Write the rows in the (non-minimal) Betti tables of Prym-canonical nodal curves as linear combinations of the rows in Pascal's triangle:

$\boldsymbol{g = 10}$

	0	1	2	3	4	5	6	7
0:	1	-	-	-	-	-	-	-
1:	-	18	55	75	54	20	3	-
2:	-	13	75	180	230	165	63	10
Σ:	1	31	130	255	284	185	66	10

=	1							·1
+	1	1						·2
+	1	2	1					·3
+	1	3	3	1				·4
+	1	4	6	4	1			·5
+	1	5	10	10	5	1		·3
	1	6	15	20	15	6	1	

Betti tables of Prym-canonical nodal curves

$g = 12$

	0	1	2	3	4	5	6	7	8	9
0:	1	-	-	-	-	-	-	-	-	-
1:	-	33	147	315	399	315	153	42	5	-
2:	-	15	117	399	777	945	735	357	99	12
Σ:	1	48	264	714	1176	1260	888	399	104	12

1								
1	1							
1	2	1						
1	3	3	1					
1	4	6	4	1				
1	5	10	10	5	1			
1	6	15	20	15	6	1		
1	7	21	35	35	21	7	1	
1	8	28	56	70	56	28	8	1

Betti tables of Prym-canonical nodal curves

$g = 12$

	0	1	2	3	4	5	6	7	8	9
0:	1	-	-	-	-	-	-	-	-	-
1:	-	33	147	315	399	315	153	42	5	-
2:	-	15	117	399	777	945	735	357	99	12
Σ:	1	48	264	714	1176	1260	888	399	104	12

=

	1								
	1	1							
	1	2	1						
	1	3	3	1					
	1	4	6	4	1				
	1	5	10	10	5	1			
	1	6	15	20	15	6	1		
	1	7	21	35	35	21	7	1		·3
+	1	8	28	56	70	56	28	8	1	·12

Betti tables of Prym-canonical nodal curves

$g = 12$

	0	1	2	3	4	5	6	7	8	9
0:	1	-	-	-	-	-	-	-	-	-
1:	-	33	147	315	399	315	153	42	5	-
2:	-	15	117	399	777	945	735	357	99	12
Σ:	1	48	264	714	1176	1260	888	399	104	12

=	1									·1
+	1	1								·2
+	1	2	1							·3
+	1	3	3	1						·4
+	1	4	6	4	1					·5
+	1	5	10	10	5	1				·6
+	1	6	15	20	15	6	1			·7
+	1	7	21	35	35	21	7	1		·5
	1	8	28	56	70	56	28	8	1	

The structure of Prym-Green matrices: Applications

Prym-Green matrix for $g = 24$

- Number of entries: 8.616.423.582
- Amount of memory (in the SINGULAR kernel, 40 Byte/entry): 321 GB

The structure of Prym-Green matrices: Applications

Prym-Green matrix for $g = 24$

- Number of entries: 8.616.423.582
- Amount of memory
  (in the SINGULAR kernel, 40 Byte/entry): 321 GB

By making use of the structure, this can be reduced to:

- Number of values: 2.306.916

The structure of Prym-Green matrices: Applications

Prym-Green matrix for $g = 24$

- Number of entries: 8.616.423.582
- Amount of memory (in the SINGULAR kernel, 40 Byte/entry): 321 GB

By making use of the structure, this can be reduced to:

- Number of values: 2.306.916
- Amount of memory (2 Byte/value): 4.4 MB

The structure of Prym-Green matrices: Applications

Prym-Green matrix for $g = 24$

- Number of entries: 8.616.423.582
- Amount of memory (in the SINGULAR kernel, 40 Byte/entry): 321 GB

By making use of the structure, this can be reduced to:

- Number of values: 2.306.916
- Amount of memory (2 Byte/value): 4.4 MB

Algorithmic solution:

- compute resolution ✔
- save matrix ✔
- reduce matrix

The structure of Prym-Green matrices: Applications

Prym-Green matrix for $g = 24$

- Number of entries: 8.616.423.582
- Amount of memory (in the SINGULAR kernel, 40 Byte/entry): 321 GB

By making use of the structure, this can be reduced to:

- Number of values: 2.306.916
- Amount of memory (2 Byte/value): 4.4 MB

Algorithmic solution:

- compute resolution ✔
- save matrix ✔
- reduce matrix **?**