Enhancing the classical algorithm by Oaku for the computation of Bernstein-Sato polynomials

Jorge Martín-Morales

Department of Mathematics
University of Zaragoza

Seminar on $D$-modules
Aachen, January 8, 2008
**Basic notations**

- \( \mathbb{C} \) the field of the complex numbers.
- \( \mathbb{C}[s] \) the ring of polynomials in one variable over \( \mathbb{C} \).
- \( R_n = \mathbb{C}[x_1, \ldots, x_n] \) the ring of polynomials in \( n \) variables.
- \( D_n = \mathbb{C}[x_1, \ldots, x_n] \langle \partial_1, \ldots, \partial_n \rangle \) the ring of \( \mathbb{C} \)-linear differential operators on \( R_n \), the \( n \)-th Weyl algebra:
  \[
  \partial_i x_i = x_i \partial_i + 1
  \]
- \( D_n[s] \) the ring of polynomials in one variable over \( D_n \).
The $D_n[s]$-module $R_n[s, \frac{1}{f}] \cdot f^s$

- Let $f \in R_n$ be a non-zero polynomial.

- By $R_n[s, \frac{1}{f}]$ we denote the ring of rational functions of the form
  \[ \frac{g(x, s)}{fr} \]
  where $g(x, s) \in R_n[s] = \mathbb{C}[x_1, \ldots, x_n, s]$.

- We denote by $M = R_n[s, \frac{1}{f}] \cdot f^s$ the free $R_n[s, \frac{1}{f}]$-module of rank one generated by the symbol $f^s$.

- $R_n[s, \frac{1}{f}] \cdot f^s$ has a natural structure of left $D_n[s]$-module.

\[ \partial_i \cdot f^s = s \frac{\partial f}{\partial x_i} \frac{1}{f} \cdot f^s \in R_n[s, \frac{1}{f}] \cdot f^s \]
Theorem (Bernstein)

For every polynomial \( f \in R_n \) there exists a non-zero polynomial \( b(s) \in \mathbb{C}[s] \) and a differential operator \( P(s) \in D_n[s] \) such that

\[
P(s)f^{s+1} = b(s)f^s \quad \in \quad R_n[s, \frac{1}{f}] \cdot f^s.
\]
Theorem (Bernstein)

For every polynomial $f \in R_n$ there exists a non-zero polynomial $b(s) \in \mathbb{C}[s]$ and a differential operator $P(s) \in D_n[s]$ such that

$$P(s)f^{s+1} = b(s)f^s \in R_n[s, \frac{1}{f}] \cdot f^s.$$

Definition (Bernstein & Sato)

The set of all possible polynomials $b(s)$ satisfying the above equation is an ideal of $\mathbb{C}[s]$. The monic generator of this ideal is denoted by $b_f(s)$ and called the Bernstein-Sato polynomial of $f$. 
Now assume that

- \( f \in \mathcal{O} = \mathbb{C}\{x_1, \ldots, x_n\} \) is a convergent power series.
- \( \mathcal{D}_n \) is the ring of differential operators with coefficients in \( \mathcal{O} \).

### The local \( b \)-function

The local \( b \)-function of \( f \) is denoted by \( b_f, 0(s) \) and called the local \( b \)-function of \( f \).
Now assume that

- $f \in \mathcal{O} = \mathbb{C}\{x_1, \ldots, x_n\}$ is a convergent power series.
- $\mathcal{D}_n$ is the ring of differential operators with coefficients in $\mathcal{O}$.

**Theorem (Björk & Kashiwara)**

For every $f \in \mathcal{O}$ there exists a non-zero polynomial $b(s) \in \mathbb{C}[s]$ and a differential operator $P(s) \in \mathcal{D}_n[s]$ such that

$$P(s)f^{s+1} = b(s)f^s \in \mathcal{O}[s, \frac{1}{f}] \cdot f^s.$$
The local \( b \)-function

Now assume that
- \( f \in \mathcal{O} = \mathbb{C}\{x_1, \ldots, x_n\} \) is a convergent power series.
- \( \mathcal{D}_n \) is the ring of differential operators with coefficients in \( \mathcal{O} \).

**Theorem (Björk & Kashiwara)**

For every \( f \in \mathcal{O} \) there exists a non-zero polynomial \( b(s) \in \mathbb{C}[s] \) and a differential operator \( P(s) \in \mathcal{D}_n[s] \) such that

\[
P(s)f^{s+1} = b(s)f^s \quad \in \quad \mathcal{O}[s, \frac{1}{f}] \cdot f^s.
\]

**Definition**

The monic polynomial in \( \mathbb{C}[s] \) of lowest degree which satisfies the above equation is denoted by \( b_{f,0}(s) \) and called the local \( b \)-function of \( f \).
Some well-known properties of the $b$-function

1. The $b$-function is always a multiple of $(s + 1)$. The equality holds if and only if $f$ is smooth.
**Some well-known properties of the $b$-function**

1. The $b$-function is always a multiple of $(s + 1)$. The equality holds if and only if $f$ is smooth.

2. The Bernstein-Sato polynomial is a non-complete analytic invariant of the singularity $f = 0$. (Malgrange).

3. The set $\{ e^{2\pi i \alpha} | b^f, 0(\alpha) = 0 \}$ is a topological invariant of the singularity $f = 0$. (Kashiwara).

4. The roots of the $b$-function are negative rational numbers of the real interval $(-n, 0)$). (Kashiwara).

5. $b^f, 0(s)$ is a divisor of $b^f(\cdot)$.

6. $b^f(\cdot) = \text{lcm}_{p \in \mathbb{C}} p(\cdot)$ (Briançon-Maisonobe, see also Mebkhout-Narváez).

J. Martín-Morales (jorge@unizar.es)

Enhancing the classical algorithm by Oaku. Applications
Some well-known properties of the $b$-function

1. The $b$-function is always a multiple of $(s + 1)$. The equality holds if and only if $f$ is smooth.

2. The Bernstein-Sato polynomial is a non-complete analytic invariant of the singularity $f = 0$.

3. The set $\{e^{2\pi i\alpha} \mid b_{f,0}(\alpha) = 0\}$ is a topological invariant of the singularity $f = 0$. (Malgrange).

4. The roots of the $b$-function are negative rational numbers of the real interval $(-n, 0)$. (Kashiwara).

5. $b_{f,0}(s)$ is a divisor of $b_f(s)$. If, for instance, $f$ has 0 as its only singularity, then $b_{f,0}(s) = b_f(s)$.

6. $b_f(s) = \text{lcm}_{p \in \mathbb{C}}(b_{f,p}(s))$ (Briançon-Maisonobe, see also Mebkhout-Narváez).
Some well-known properties of the \( b \)-function

1. The \( b \)-function is always a multiple of \((s + 1)\). The equality holds if and only if \( f \) is smooth.

2. The Bernstein-Sato polynomial is a non-complete analytic invariant of the singularity \( f = 0 \).

3. The set \( \{ e^{2\pi i \alpha} \mid b_{f,0}(\alpha) = 0 \} \) is a topological invariant of the singularity \( f = 0 \). (Malgrange).

4. The roots of the \( b \)-function are negative rational numbers of the real interval \((-n, 0)\). (Kashiwara).
Some well-known properties of the $b$-function

1. The $b$-function is always a multiple of $(s + 1)$. The equality holds if and only if $f$ is smooth.

2. The Bernstein-Sato polynomial is a non-complete analytic invariant of the singularity $f = 0$.

3. The set $\left\{ e^{2\pi i \alpha} \mid b_{f,0}(\alpha) = 0 \right\}$ is a topological invariant of the singularity $f = 0$. (Malgrange).

4. The roots of the $b$-function are negative rational numbers of the real interval $(-n, 0)$. (Kashiwara).

5. $b_{f,0}(s)$ is a divisor of $b_f(s)$. If, for instance, $f$ has 0 as its only singularity, then $b_{f,0}(s) = b_f(s)$. 

J. Martín-Morales (jorge@unizar.es)
Some well-known properties of the $b$-function

1. The $b$-function is always a multiple of $(s + 1)$. The equality holds if and only if $f$ is smooth.

2. The Bernstein-Sato polynomial is a non-complete analytic invariant of the singularity $f = 0$.

3. The set $\{ e^{2\pi i \alpha} \mid b_{f,0}(\alpha) = 0 \}$ is a topological invariant of the singularity $f = 0$. (Malgrange).

4. The roots of the $b$-function are negative rational numbers of the real interval $(-n, 0)$. (Kashiwara).

5. $b_{f,0}(s)$ is a divisor of $b_f(s)$. If, for instance, $f$ has 0 as its only singularity, then $b_{f,0}(s) = b_f(s)$.

6. $b_f(s) = \text{lcm}_{p \in \mathbb{C}^n}(b_{f,p}(s))$ (Briançon-Maisonobe, see also Mebkhout-Narváez).
Algorithms for computing the $b$-function

Global $b$-function.
Isolated case: use the algorithm implemented by Mathias Schulze in Singular for computing the local $b$-functions and then apply the formula $b_f(s) = \text{lcm}_{p \in C} (b_f, p(s))$.
Non-isolated case: use the algorithm by Oaku and Takayama based on Gröbner bases in the ring of differential operators.

Local $b$-function.
Isolated case: use the algorithm implemented by Mathias Schulze in Singular for computing the local $b$-function.
Non-isolated case: use the algorithm by Oaku and Takayama based on Gröbner bases in a local ring of differential operators.

Enhancing the classical algorithm by Oaku. Applications
Global $b$-function.

- Isolated case: use the algorithm implemented by Mathias Schulze in SINGULAR for computing the local $b$-functions and then apply the formula $b_f(s) = \text{lcm}_{p \in \mathbb{C}^n}(b_f, p(s))$.

Non-isolated case: use the algorithm by Oaku and Takayama based on Gröbner bases in the ring of differential operators.
Global $b$-function.

- Isolated case: use the algorithm implemented by Mathias Schulze in SINGULAR for computing the local $b$-functions and then apply the formula $b_f(s) = \text{lcm}_{p \in \mathbb{C}^n}(b_f, p(s))$.

- Non-isolated case: use the algorithm by Oaku and Takayama based on Gröbner bases in the ring of differential operators.
Global $b$-function.

- Isolated case: use the algorithm implemented by Mathias Schulze in *SINGULAR* for computing the local $b$-functions and then apply the formula $b_f(s) = \text{lcm}_{p \in \mathbb{C}^n}(b_f, p(s))$.

- Non-isolated case: use the algorithm by Oaku and Takayama based on Gröbner bases in the ring of differential operators.

Local $b$-function.

- Isolated case: use the algorithm implemented by Mathias Schulze in *SINGULAR* for computing the local $b$-function.
**Algorithms for computing the $b$-function**

1. **Global $b$-function.**
   - **Isolated case:** use the algorithm implemented by Mathias Schulze in *SINGULAR* for computing the local $b$-functions and then apply the formula $b_f(s) = \text{lcm}_{p \in \mathbb{C}^n}(b_f, p(s))$.
   - **Non-isolated case:** use the algorithm by Oaku and Takayama based on Gröbner bases in the ring of differential operators.

2. **Local $b$-function.**
   - **Isolated case:** use the algorithm implemented by Mathias Schulze in *SINGULAR* for computing the local $b$-function.
   - **Non-isolated case:** use the algorithm by Oaku and Takayama based on Gröbner bases in a *local* ring of differential operators.
Another idea for computing the $b$-function

1. Obtain an upper bound for $b_f(s)$: find $B(s) \in \mathbb{C}[s]$ such that $b_f(s)$ divides $B(s)$.

   $$B(s) = d \prod_{i=1}^{m_i} (s - \alpha_i)^{m_i}.$$ 

2. Check whether $\alpha_i$ is a root of the $b$-function.

3. Compute the multiplicity of $\alpha_i$ as a root of $b_f(s)$.

Remark: There are some well-known methods to obtain such $B(s)$: Resolution of Singularities.

We need two algorithms.
Another idea for computing the \( b \)-function

1. Obtain an upper bound for \( b_f(s) \): find \( B(s) \in \mathbb{C}[s] \) such that \( b_f(s) \) divides \( B(s) \).

\[
B(s) = \prod_{i=1}^{d} (s - \alpha_i)^{m_i}.
\]

Remark: There are some well-known methods to obtain such \( B(s) \): Resolution of Singularities.
Another idea for computing the $b$-function

1. Obtain an upper bound for $b_f(s)$: find $B(s) \in \mathbb{C}[s]$ such that $b_f(s)$ divides $B(s)$.

   \[ B(s) = \prod_{i=1}^{d} (s - \alpha_i)^{m_i}. \]

2. Check whether $\alpha_i$ is a root of the $b$-function.

Remark

There are some well-known methods to obtain such $B(s)$: Resolution of Singularities. We need two algorithms.
Another idea for computing the $b$-function

1. Obtain an upper bound for $b_f(s)$: find $B(s) \in \mathbb{C}[s]$ such that $b_f(s)$ divides $B(s)$.

$$B(s) = \prod_{i=1}^{d} (s - \alpha_i)^{m_i}.$$ 

2. Check whether $\alpha_i$ is a root of the $b$-function.

3. Compute the multiplicity of $\alpha$ as a root of $b_f(s)$. 

Remark: There are some well-known methods to obtain such $B(s)$: Resolution of Singularities. We need two algorithms.

J. Martín-Morales (jorge@unizar.es) Enhancing the classical algorithm by Oaku. Applications
Another idea for computing the $b$-function

1. Obtain an upper bound for $b_f(s)$: find $B(s) \in \mathbb{C}[s]$ such that $b_f(s)$ divides $B(s)$.

$$B(s) = \prod_{i=1}^{d} (s - \alpha_i)^{m_i}.$$ 

2. Check whether $\alpha_i$ is a root of the $b$-function.

3. Compute the multiplicity of $\alpha$ as a root of $b_f(s)$.

Remark

- There are some well-known methods to obtain such $B(s)$: Resolution of Singularities.
- We need two algorithms.
The main trick

By definition, \( (\text{ann}_{D_n}[s](f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle \).
THE MAIN TRICK

- By definition, \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle\).
- \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] + \langle s + \alpha \rangle = \langle b_f(s), s + \alpha \rangle\)
The main trick

- By definition, \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle\).
- \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] + \langle s + \alpha \rangle = \langle b_f(s), s + \alpha \rangle\)
The main trick

- By definition, $(\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle$.
- $(\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] + \langle s + \alpha \rangle = \langle b_f(s), s + \alpha \rangle$

**Proposition**

$$(\text{ann}_{D_n[s]}(f^s) + \langle f, s + \alpha \rangle) \cap \mathbb{C}[s] = \langle b_f(s), s + \alpha \rangle$$

$$= \begin{cases} 
\langle s + \alpha \rangle & \text{si } b_f(-\alpha) = 0 \\
\mathbb{C}[s] & \text{otherwise}
\end{cases}$$
The main trick

- By definition, \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle\).

- \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] + \langle s + \alpha \rangle = \langle b_f(s), s + \alpha \rangle\)

**Proposition**

\[
(\text{ann}_{D_n[s]}(f^s) + \langle f, s + \alpha \rangle) \cap \mathbb{C}[s] = \langle b_f(s), s + \alpha \rangle
\]

\[
= \begin{cases} 
\langle s + \alpha \rangle & \text{if } b_f(-\alpha) = 0 \\
\mathbb{C}[s] & \text{otherwise}
\end{cases}
\]

**Corollary**

The following conditions are equivalent:

1. \(\alpha \in \mathbb{Q}\) is a root of \(b_f(-s)\).
2. \(\text{ann}_{D_n[s]}(f^s) + \langle f, s + \alpha \rangle \neq D_n[s]\).
3. \(\text{ann}_{D_n[s]}(f^s)|_{s=-\alpha} + \langle f \rangle \neq D\).
Algorithm 1 (check whether $\alpha \in \mathbb{Q}$ is a root of the $b$-function)

Input: $I = \text{ann}_{D_n[s]}(f^s)$, $f$ a polynomial in $R_n$, $\alpha \in \mathbb{Q}$;
Output: true if $\alpha$ is a root of $b_f(-s)$, false otherwise;

1. $J := I|_{s=-\alpha} + \langle f \rangle$;  \hspace{1cm} \triangleright J \subseteq D_n$
2. $G$ a reduced Gröbner basis of $J$ w.r.t. any term ordering;
3. if $G \neq \{1\}$ then
   return true
else
   return false
end if
What about the multiplicity?

By definition, \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle\).

\((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] + \langle q(s) \rangle = \langle b_f(s), q(s) \rangle, \quad q(s) \in \mathbb{C}[s]\)
What about the multiplicity?

- By definition, \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle\).
- \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] + \langle q(s) \rangle = \langle b_f(s), q(s) \rangle, \; q(s) \in \mathbb{C}[s]\)

**Proposition**

\[
(\text{ann}_{D_n[s]}(f^s) + \langle f, q(s) \rangle) \cap \mathbb{C}[s] = \langle b_f(s), q(s) \rangle \\
= \langle \text{gcd}(b_f(s), q(s)) \rangle
\]
What about the multiplicity?

- By definition, \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] = \langle b_f(s) \rangle\).

- \((\text{ann}_{D_n[s]}(f^s) + \langle f \rangle) \cap \mathbb{C}[s] + \langle q(s) \rangle = \langle b_f(s), q(s) \rangle, \quad q(s) \in \mathbb{C}[s]\)

**Proposition**

\[
(\text{ann}_{D_n[s]}(f^s) + \langle f, q(s) \rangle) \cap \mathbb{C}[s] = \langle b_f(s), q(s) \rangle = \langle \gcd(b_f(s), q(s)) \rangle
\]

**Corollary**

- \(m_\alpha\) the multiplicity of \(\alpha\) as a root of \(b_f(-s)\).
- \(J_i = \text{ann}_{D_n[s]}(f^s) + \langle f, (s + \alpha)^{i+1} \rangle \subseteq D_n[s]\).

The following conditions are equivalent:

1. \(m_\alpha > i\).
2. \((s + \alpha)^i \notin J_i\).
**Algorithm 2**

**Algorithm 2** (compute the multiplicity of $\alpha$ as a root of $b_f(-s)$)

Input: $I = \text{ann}_{D_n[s]}(f^s)$, $f$ a polynomial in $R_n$, $\alpha$ in $\mathbb{Q}$;  
Output: $m_\alpha$, the multiplicity of $\alpha$ as a root of $b_f(-s)$;

for $i = 0$ to $n$ do  
1. $J := I + \langle f, (s + \alpha)^{i+1} \rangle$; \hspace{1cm} $\triangleright J_i \subseteq D_n[s]$  
2. $G$ a reduced Gröbner basis of $J$ w.r.t. any term ordering;  
3. $r$ normal form of $(s + \alpha)^i$ with respect to $G$;  
4. if $r = 0$ then \hspace{1cm} $\triangleright r = 0 \iff (s + \alpha)^i \in J_i$  
   $m_\alpha = i$; \hspace{1cm} break \hspace{1cm} $\triangleright$ leave the for block  
end if
end for
return $m_\alpha$
Remember the idea for computing $b_f(s)$

1. Obtain an upper bound for $b_f(s)$: find $B(s) \in \mathbb{C}[s]$ such that $b_f(s)$ divides $B(s)$.

\[ B(s) = \prod_{i=1}^{d} (s - \alpha_i)^{m_i}. \]

2. Check whether $\alpha_i$ is a root of the $b$-function.

3. Compute its multiplicity $m_i$.

What about the first step?
Let us see the following applications:

1. Computations of the $b$-functions via embedded resolutions.
2. Computations of the $b$-function of deformation of singularities.
3. An algorithm for computing the minimal integral root of $b_f(s)$ without computing the whole Bernstein-Sato polynomial.
Let $f \in \mathcal{O}$ be a convergent power series, $f : \Delta \subseteq \mathbb{C}^n \rightarrow \mathbb{C}$.

Assume that $f(0) = 0$, otherwise $b_{f,0}(s) = 1$.

Let $\varphi : Y \rightarrow \Delta$ be an embedded resolution of $\{f = 0\}$.

If $F = f \circ \varphi$, then $F^{-1}(0)$ is a normal crossing divisor.
Let \( f \in \mathcal{O} \) be a convergent power series, \( f : \Delta \subseteq \mathbb{C}^n \rightarrow \mathbb{C} \).

Assume that \( f(0) = 0 \), otherwise \( b_{f,0}(s) = 1 \).

Let \( \varphi : Y \rightarrow \Delta \) be an embedded resolution of \( \{ f = 0 \} \).

If \( F = f \circ \varphi \), then \( F^{-1}(0) \) is a normal crossing divisor.

**Theorem (Kashiwara).**

There exists an integer \( k \geq 0 \) such that \( b_f(s) \) is a divisor of the product \( b_F(s) b_F(s + 1) \cdots b_F(s + k) \).
Let us consider $f = y^2 - x^3 \in \mathbb{C}\{x, y\}$. 
Example

Let us consider \( f = y^2 - x^3 \in \mathbb{C}\{x, y\} \).

\[
\varphi^{-1}(X) \subseteq Y
\]

\[
X \subseteq \mathbb{C}^2
\]

\[
0 \xrightarrow{\varphi} E_3 \rightarrow E_1 \rightarrow E_2 \rightarrow E_4 \rightarrow 6
\]

2 3 1

\[
\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1, \frac{7}{6}, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, \frac{11}{6}
\]

Using algorithms 1 and 2, we have proved that the numbers in red are the roots of \( b_f(s) \), all of them with multiplicity one.
**Example**

- Let us consider $f = y^2 - x^3 \in \mathbb{C}\{x, y\}$.

- From Kashiwara, the possible roots of $b_f(-s)$ are:
  
  $\begin{align*}
  &1, 1, 1, 2, 5, 7, 4, 3, 5, 11 \\
  &\bar{6}', \bar{3}', \bar{2}', \bar{3}', \bar{6}', \bar{1}', \bar{6}', \bar{3}', \bar{2}', \bar{3}', \bar{6}'.
  \end{align*}$

\[ X \subseteq \mathbb{C}^2 \]

\[ \varphi^{-1}(X) \subseteq Y \]

\[ \varphi \]

\[ E_1 \quad E_2 \quad E_3 \quad E_4 \]

\[ 2 \quad 3 \quad 1 \quad 6 \]
Let us consider \( f = y^2 - x^3 \in \mathbb{C}\{x, y\} \).

From Kashiwara, the possible roots of \( b_f(-s) \) are:

\[
1, 1, 1, 2, 5, 7, 4, 3, 5, 11, \overline{6}, \overline{3}, \overline{2}, \overline{3}, \overline{6}, \overline{1}, \overline{6}, \overline{3}, \overline{2}, \overline{3}, \overline{6}.
\]

Using algorithms 1 and 2, we have proved that the numbers in red are the roots of \( b_f(s) \), all of them with multiplicity one.
Using this method we have computed the $b$-function of $f = (xz + y)(x^4 + y^5 + xy^4)$ which is a non-isolated singularity.
Let $f, g$ be two topologically equivalent singularities.
Let $f, g$ be two topologically equivalent singularities.

Assume that $b_f(s)$ is known.
Let $f, g$ be two topologically equivalent singularities.

Assume that $b_f(s)$ is known.

Since the set $\{e^{2\pi i \alpha} \mid b_f(\alpha) = 0\}$ is a topological invariant of the singularity $f = 0$ and every root belongs to $(-n, 0)$, one can find an upper bound for $b_g(s)$. 
Let $f, g$ be two topologically equivalent singularities.

Assume that $b_f(s)$ is known.

Since the set $\{e^{2\pi i \alpha} \mid b_f(\alpha) = 0\}$ is a topological invariant of the singularity $f = 0$ and every root belongs to $(-n, 0)$, one can find an upper bound for $b_g(s)$.

Then we use algorithms 1 and 2 for computing $b_g(s)$. 
Let $f = x^4 + y^5$ and $g = x^4 + y^5 + xy^4$. 

The possible roots of $b^g(-s)$ are:


Using algorithms 1 and 2, we have proved that the numbers in red are the roots of $b^g(-s)$, all of them with multiplicity one.
Example

- Let \( f = x^4 + y^5 \) and \( g = x^4 + y^5 + xy^4 \).
- \( f \) and \( g \) are topologically equivalent because they have the same Puiseux pairs.
**Example**

- Let $f = x^4 + y^5$ and $g = x^4 + y^5 + xy^4$.
- $f$ and $g$ are topologically equivalent because they have the same Puiseux pairs.
- The following numbers are the roots of $b_f(-s)$, all of them with multiplicity one.

\[
\frac{9}{20}, \frac{13}{20}, \frac{7}{10}, \frac{17}{20}, \frac{9}{10}, \frac{19}{20}, \frac{21}{10}, \frac{11}{20}, \frac{23}{10}, \frac{13}{20}, \frac{27}{20}, \frac{31}{20}
\]
Example

- Let $f = x^4 + y^5$ and $g = x^4 + y^5 + xy^4$.
- $f$ and $g$ are topologically equivalent because they have the same Puiseux pairs.
- The following numbers are the roots of $b_f(-s)$, all of them with multiplicity one.

\[
\frac{9}{20}, \frac{13}{20}, \frac{7}{10}, \frac{17}{20}, \frac{9}{10}, \frac{19}{20}, \frac{21}{10}, \frac{11}{20}, \frac{23}{10}, \frac{13}{20}, \frac{27}{20}, \frac{31}{20}
\]

- The possible roots of $b_g(-s)$ are:

\[
\frac{9}{20}, \frac{13}{20}, \frac{7}{10}, \frac{17}{20}, \frac{9}{10}, \frac{19}{20}, \frac{21}{10}, \frac{11}{20}, \frac{23}{10}, \frac{13}{20}, \frac{27}{20}, \frac{31}{20}, \\
\frac{29}{20}, \frac{33}{20}, \frac{17}{10}, \frac{37}{20}, \frac{19}{10}, \frac{39}{20}, \frac{1}{20}, \frac{1}{10}, \frac{3}{10}, \frac{7}{20}, \frac{11}{20}
\]
**Example**

- Let \( f = x^4 + y^5 \) and \( g = x^4 + y^5 + xy^4 \).
- \( f \) and \( g \) are topologically equivalent because they have the same Puiseux pairs.
- The following numbers are the roots of \( b_f(-s) \), all of them with multiplicity one.

\[
\frac{9}{20}, \frac{13}{20}, \frac{7}{10}, \frac{17}{20}, \frac{9}{10}, \frac{19}{20}, \frac{21}{20}, \frac{11}{10}, \frac{13}{10}, \frac{27}{20}, \frac{31}{20}
\]

- The possible roots of \( b_g(-s) \) are:

\[
\frac{9}{20}, \frac{13}{20}, \frac{7}{10}, \frac{17}{20}, \frac{9}{10}, \frac{19}{20}, \frac{21}{20}, \frac{11}{10}, \frac{13}{10}, \frac{27}{20}, \frac{31}{20}, \\
\frac{29}{20}, \frac{33}{20}, \frac{17}{10}, \frac{37}{20}, \frac{19}{10}, \frac{39}{20}, \frac{1}{20}, \frac{1}{10}, \frac{3}{10}, \frac{7}{20}, \frac{11}{20}
\]

- Using algorithms 1 and 2, we have proved that the numbers in red are the roots of \( b_g(-s) \), all of them with multiplicity one.
Using this method we have computed the Bernstein polynomial for \( g = z^4 + x^6 y^5 + x^5 y^4 z \).
We chose \( f = z^4 + x^6 y^5 \) which is topologically equivalent to \( g \).
THE MINIMAL INTEGRAL ROOT OF $b_f(s)$

Example

Let us consider the following example:
THE MINIMAL INTEGRAL ROOT OF $b_f(s)$

Example

Let us consider the following example:

$$A = \begin{pmatrix}
    x_1 & x_2 & x_3 & x_4 \\
    x_5 & x_6 & x_7 & x_8 \\
    x_9 & x_{10} & x_{11} & x_{12}
\end{pmatrix}$$
Let us consider the following example:

\[
A = \begin{pmatrix}
x_1 & x_2 & x_3 & x_4 \\
x_5 & x_6 & x_7 & x_8 \\
x_9 & x_{10} & x_{11} & x_{12}
\end{pmatrix}
\]

- \( \Delta_i \): determinant of the minor resulting from deleting the \( i \)-th column of \( A \), \( i = 1, 2, 3, 4 \).
THE MINIMAL INTEGRAL ROOT OF $b_f(s)$

Example

Let us consider the following example:

\[ A = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \end{pmatrix} \]

- $\Delta_i$: determinant of the minor resulting from deleting the $i$-th column of $A$, $i = 1, 2, 3, 4$.

- $f = \Delta_1 \Delta_2 \Delta_3 \Delta_4 \in \mathbb{C}[x_1, \ldots, x_{12}]$. 
Let us consider the following example:

$$A = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \end{pmatrix}$$

- $\Delta_i$ determinant of the minor resulting from deleting the $i$-th column of $A$, $i = 1, 2, 3, 4$.
- $f = \Delta_1 \Delta_2 \Delta_3 \Delta_4 \in \mathbb{C}[x_1, \ldots, x_{12}]$.

From Kashiwara, the possible integral roots of $b_f(-s)$ are

$$11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.$$ 

Using the algorithm 1, we have proved that the minimal integral root of $b_f(s)$ is $-1$. 
At the moment the SINGULAR library dmod.lib for algebraic $D$-modules contains the following main procedures:

- **Sannfs**: computes a system of generators of $\text{ann}_{D[s]}(f^s)$.
- **Sannfslog**: computes a system of generators of $\text{ann}^{(1)}_{D[s]}(f^s)$.
- **SannfsParam**: computes a system of generators of $\text{ann}_{D[s]}(f^s)$ when $f$ has parameters.
- **checkRoot**
- **annfs**
- **operator**: computes $P(s)$ such that $P(s)f^{s+1} = b_f(s)f^s$.
- **isHolonomic**: checks whether a module given by a presentation is holonomic.
Thank you very much!

J. Martín-Morales (jorge@unizar.es)

Department of Mathematics
University of Zaragoza

Seminar on $D$-modules
Aachen, January 8, 2008