SoftwareX 9 (2019) 35–38

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Automating the calculation of the Hilbert–Kunz multiplicity and F-signature

Gabriel Johnson [a], Sandra Spiroff [b,*,1]

[a] Department of Computer Science, University of Mississippi, Oxford, MS 38677, United States
[b] Department of Mathematics, University of Mississippi, Oxford, MS 38677, United States

ARTICLE INFO

Article history:
Received 20 September 2018
Received in revised form 4 December 2018
Accepted 4 December 2018

MSC:
68-04
13-04
15A39

Keywords:
Hilbert–Kunz multiplicity
F-signature
Intersection algebras

ABSTRACT

The Hilbert–Kunz multiplicity and F-signature are important invariants for researchers in commutative algebra and algebraic geometry. We provide software, and describe the automation, for the calculations of the two invariants in the case of intersection algebras over polynomial rings.

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_153
Legal Code License	BSD 3-Clause
Code versioning system used	Git
Software code languages, tools, and services used	Common Lisp, Bash
Compilation requirements, operating environments & dependencies	Clozure Common Lisp, Bash/Cygwin, Sed, Mathematica (optional)
If available Link to developer documentation/manual	
Support email for questions	gsjohnso@go.olemiss.edu

1. Motivation and significance

The Hilbert–Kunz multiplicity, along with the F-signature, has much importance in the related fields of commutative algebra and algebraic geometry, specifically, in characteristic $p > 0$ methods, but is notoriously difficult to compute in practice (see, for example, [1]). In particular, very few examples exist in the literature where both values are known simultaneously. This state of affairs motivated the work of Enescu and Spiroff, who calculated the two invariants, as well as the Hilbert–Samuel multiplicity and divisor class group, for certain classes of intersection algebras [2]. In their (toric) setting, the Hilbert–Kunz multiplicity and F-signature can be found using volumes of polytopes [2, Proposition 4.2], [3, Theorem 2.2], [4, Theorem 3.2.3]. Thus, their calculation, being combinatorial in nature, lends itself perfectly to computer automation. One might expect to be able to obtain general formulæ for the invariants, however, a significant hurdle is the lack of a usable description of the unique Hilbert basis elements in terms of the parameters.

The **intersection algebra** of a commutative ring R in terms of ideals I and J is $\mathcal{B} = \mathcal{B}_R(I, J) = \bigoplus_{r,s\in\mathbb{Z}_{\geq 0}}(I^r \cap J^s)$. In particular, when $R = \Bbbk[x_1, \ldots, x_n]$, for a field $\Bbbk$, and $I = (x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n})$ and $J = (x_1^{b_1}x_2^{b_2}\cdots x_n^{b_n})$, with $a_i, b_i \in \mathbb{Z}_{\geq 0}$, the ordered pairs (b_i, a_i) partition the first quadrant of the plane into a fan made up of pointed rational cones, from which monoids, and Hilbert bases are obtained.

* Corresponding author.
E-mail addresses: gsjohnso@go.olemiss.edu (G. Johnson), spiroff@go.olemiss.edu (S. Spiroff).

1 S. Spiroff was supported by Simons Foundation, United States Collaboration Grant #245926.

https://doi.org/10.1016/j.softx.2018.12.002
2352-7110/

While for general $n \in \mathbb{N}$ and $\mathbf{a} = (a_1, \ldots, a_n)$, $\mathbf{b} = (b_1, \ldots, b_n)$ no formula for the Hilbert–Kunz multiplicity $e_{HK}(\mathcal{B})$ or the F-signature $s(\mathcal{B})$ is known, Enescu and Spiroff have an algorithm to compute both, for any specific numerical entries [2, §5]. This algorithm has been automated by Johnson. Thus, while researchers have in the past relied on a bound of the invariants, or only had information about one of the two, they may now know the exact value of both for rings in this class by a simple extraction of numerical data.

Intersection algebras overlap with classes of objects in algebraic geometry. For example, when $R = \mathsf{k}[x]$, $\mathcal{B}_R((x^{a_1}), (x))$ is isomorphic to the rational normal scroll $S = \mathsf{k}[T, xT, xyT, yT, x^{-1}yT, x^{-2}yT, \ldots, x^{-(a_1-1)}yT]$ [5, Example 3.5], and $\mathcal{B}_R((x), (x))$ is a Segre product [6, §3], [2, Proposition 1.12], i.e., the homogeneous coordinate ring for the Segre embedding $\mathbb{P}^1 \times \mathbb{P}^1 \to \mathbb{P}^3$. For more details on intersection algebras, especially background material, see [7, §2] and [8], and for the complete set of results obtained by Enescu and Spiroff, see [2]. References for geometric results are [1,5,6]. Throughout the paper, R will be a polynomial ring over a field k and I, J principal monomial ideals. Since the numerical invariants in our case are given by volumes, they are independent of the characteristic of k.

2. Software description

To calculate the Hilbert–Kunz multiplicity and F-signature for $\mathcal{B}_R(I, J) = \bigoplus_{r,s\in\mathbb{Z}_{\geq 0}}(I^r \cap J^s)$, where $R = \mathsf{k}[x_1, \ldots, x_n]$, $I = (x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n})$, and $J = (x_1^{b_1}x_2^{b_2}\cdots x_n^{b_n})$, with $a_i, b_i \in \mathbb{N}$, the user extracts (and if necessary, permutes) the exponents into two strings of *fan ordered* positive[2] integers, $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, meaning $a_i/b_i \geq a_{i+1}/b_{i+1}$ for all $1 \leq i < n$. Then the user commands are *user directory*\$ `./calculate-integral` $a_1\ a_2 \cdots a_n\ b_1\ b_2 \cdots b_n$ and *user directory*\$ `./inequalities` $a_1\ a_2 \cdots a_n\ b_1\ b_2 \cdots b_n$.

The command line interface can be run on a Unix shell. If the first command is invoked, then the program displays the exact values of the two invariants for $\mathcal{B}_R(I, J)$, necessarily rational numbers [6, Theorem 2.1], on two lines and terminates, as shown below.

```
Hilbert--Kunz Multiplicity        = p1/q1
F-Signature                       = p2/q2
```

Otherwise, the second command displays the *Mathematica* code used to calculate these values. See Example 3.1.

2.1. Software architecture and functionalities

Three main technologies are used.

1. The Bash shell. This collects the arguments from the user and invokes the program itself.

2. Clozure Common Lisp. This is the programming language we used to calculate the set of inequalities bounding the region in question. It provides a large number of list-processing operators as well as highly flexible looping constructs, making it well suited for dealing with the points of indefinite dimension involved in this project.

3. *Mathematica* [9]. This is used to calculate the final values of the integrals that give the Hilbert–Kunz multiplicity and F-signature, in any number of dimensions. The integration features of *Mathematica* are particularly well suited to this project, where the region of integration is defined by a set of inequalities in any number of dimensions. Other tools capable of solving integrals tend to require a function rather than a set of inequalities to define the area of integration, and limit the number of dimensions to two or three.

The software package consists of a subdirectory called ccl containing Clozure Common Lisp itself, another subdirectory called src containing the Lisp source code as well as a couple of auxiliary bash scripts, and three main bash scripts: setup, inequalities, and calculate-integral. The first, setup, must be run before attempting to run the others. It has Clozure compile the source code into an image file (effectively a DLL; in Lisp, all programs are essentially implemented as a DLL loaded by the language kernel with a specified top-level function to run instead of a REPL) called hkm.image. Once the hkm.image exists in the same directory as the other files, calculate-integral can be run as specified above.

All the files and subdirectories must occupy the same root directory. Further, for users with *Mathematica*, the program also assumes that the *Mathematica* kernel, called MathKernel or Wolfram, exists in a specified location depending on the operating system.

2.2. Implementation details

Below is a brief overview of the algorithm used in our program for either of the two commands. Once the arguments are collected[3] from the command line, all steps but the integration for the volumes are done entirely in Lisp.

Step 1. Our program first finds the Hilbert set for $\mathcal{B}_R(I, J)$, [2, §1], [7]. Associated to the pairs (b_i, a_i), there are cones C_i and C, and monoids Q_i, and the unique Hilbert basis $\mathcal{H}_i$ for each Q_i is found via the algorithm given in [10, Algorithm 2.4]. The **Hilbert set** is $\mathcal{H} = \cup_{i=0,\ldots,n}\mathcal{H}_i$.

Step 2. It finds the set $\mathcal{G} = \{(v, \mathbf{t}(v)) : v \in \mathcal{H}\}$, where $\mathbf{t}(v) = (\max(a_i r, b_i s))_{i=1,\ldots,n}$ for $v = (r, s) \in \mathcal{H}$. Each pair in the Hilbert set will be rewritten as an $(n + 2)$-tuple. The extra n coordinates z_i are determined according to the formula $z_i(x, y) = \max(a_i x, b_i y)$, corresponding to the intersection of monomial ideals.

Step 3. It determines the first set of inequalities bounding a region in $n + 2$ dimensions. If the individual coordinates of each element of $\mathcal{G}$ are labeled as $(x, y, z_1, z_2, \ldots, z_n)$, then an initial set of inequalities is derived by a predetermined template:

$$\bigwedge_{i=1}^{n} 0 \leq a_i x \leq z_i \wedge 0 \leq b_i y \leq z_i.$$

From these initial inequalities and $\mathcal{G}$, further inequalities must be derived, and the conjunction of all these inequalities will produce the region we seek: an element $(p, q, r_1, r_2, \ldots, r_n) \in \mathcal{G}$ generates a set of further inequalities by the following function:

$$\begin{aligned}&\text{extraInequality}(p, q, r_1, r_2, \ldots, r_n) = \\ &\quad \neg\text{initialInequalities}(x-p, y-q, z_1-r_1, z_2-r_2, \ldots, z_n-r_n),\end{aligned}$$

where initialInequalities is a function that accepts an element of $\mathcal{G}$ and returns true if and only if the given point satisfies all the initial inequalities. The program repeats this process for every element of $\mathcal{G}$. Then, it joins all the sets of inequalities by conjunction, producing our region, the volume of which is the Hilbert–Kunz multiplicity [2, Proposition 4.2], [3, Theorem 2.2].

Step 4. It determines the second set of inequalities, for the F-signature, bounding another region in $n + 2$ dimensions, using the formula

$$\bigwedge_{i=1}^{n} a_i x \leq z_i \leq 1 + a_i x \wedge b_i y \leq z_i \leq 1 + b_i y.$$

This can be found with a simple for loop. The F-signature is the volume of this polytope [4, Theorem 3.2.3].

Output. If the `./inequalities` command is invoked, then the program shows the *Mathematica* code used in the calculations,

[2] The case of non-negative integers can be addressed within the scope of positive integers. See [2, Proposition 1.6].

[3] Recall that $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ must be entered in fan order. See p. 2.

which includes both sets of inequalities bounding the regions determined in steps 3 and 4, and terminates. If the `./calculate-integral` command is used, then the program passes this code to *Mathematica*, which calculates, by integration, the volume of each region. Sed is used to format the output readably.

Remark 2.1. The `./inequalities` command allows one to integrate or analyze the invariants separately or via another program, especially if he/she does not have a local installation of *Mathematica*, a proprietary software. As evinced in the examples below, the running time in higher dimensions for the `./calculate-integral` command can significantly increase depending upon the complexity of the geometry involved. The bottleneck in these cases is the integration; the `./inequalities` command finished almost instantaneously even in the most complex cases tested. It can also be used for 3D printing.

3. Illustrative examples

Example 3.1 (*Volumes.*)**.** For $\mathcal{B}_R(I, J) = \mathcal{B}_R((x^3), (x^2))$, where $R = \Bbbk[x]$, the command `./inequalities 3 2` produces the following output.

```
   Integrate[Boole[ 0 <= x && 0 <= y && 3x <= z1 && 2y
<= z1 &&
((z1 < 3x + 1 || z1 < 2y + 1)) && ((x < 1 || y < 2 || z1
< 3x + 1))
&& ((y < 1 || z1 < 3x + 2)) && ((x < 1 || z1 < 2y + 3))
&&
((x < 1 || y < 1 || z1 < 2y + 1)) && ((x < 2 || y < 3))],
{x, 0, 2000}, {y, 0, 2000}, {z1, 0, 2000}]

   Integrate[Boole[3x <= z1 < 1 + 3x && 2y <= z1 < 1 +
2y],
{x, 0, 1}, {y, 0, 1}, {z1, 0, 30}]
```

The solids bounded by the inequalities are graphed below using *Mathematica*. The Hilbert–Kunz multiplicity and F-signature of $\mathcal{B}_R((x^3), (x^2))$ are the volumes of the solids, respectively, as per [2] (see Figs. 1 and 2). Their exact values are obtained via the command `./calculate-inequalities 3 2`. Moreover, to provide additional clarity, we note that $\mathcal{B}_R((x^3), (x^2)) \cong \Bbbk[x, T_1, T_2, T_3, T_4, T_5]/\mathscr{I}$ where $\mathscr{I} = (T_1T_3 - T_2^2, T_1T_4 - xT_2, T_1T_5 - x^2T_4, T_2T_4 - xT_3, T_2T_5 - xT_4^2, T_3T_5 - T_4^3)$, as described in [10, Example 3.14].

Example 3.2 (*Fan Order; Running Times*)**.** When $R = \Bbbk[x, y, z]$, $I = (xy^6z^5)$, and $J = (x^2y^4z^7)$, the commands in our program are necessarily implemented in fan order: `2 7 4 1 5 6`. The running time of the `./calculate-integral` command, whose output for $\mathcal{B}_R(I, J)$ is displayed below, was approximately three and a half hours, but `./inequalities` executed almost instantaneously.

```
Hilbert--Kunz Multiplicity   = 1874881259711/391184640000
F-Signature                    = 27251293/1564738560
```

As a related example, fan order for $\mathcal{B}_R(I, I)$ is obtained by any sequence $a_i\ a_j\ a_k\ a_i\ a_j\ a_k$ and `./calculate-integral 1 6 5 1 6 5`, for example, executes in a matter of seconds because of the symmetry of the associated geometric region. We obtain $e_{HK}(\mathcal{B}_R(I, I)) = e_{HK}(\mathcal{B}_R((xy^6z^5), (xy^6z^5))) = \frac{1633}{864}$ and $s(\mathcal{B}_R(I, I)) = \frac{95}{864}$, and note that the sum of the two invariants equals two.

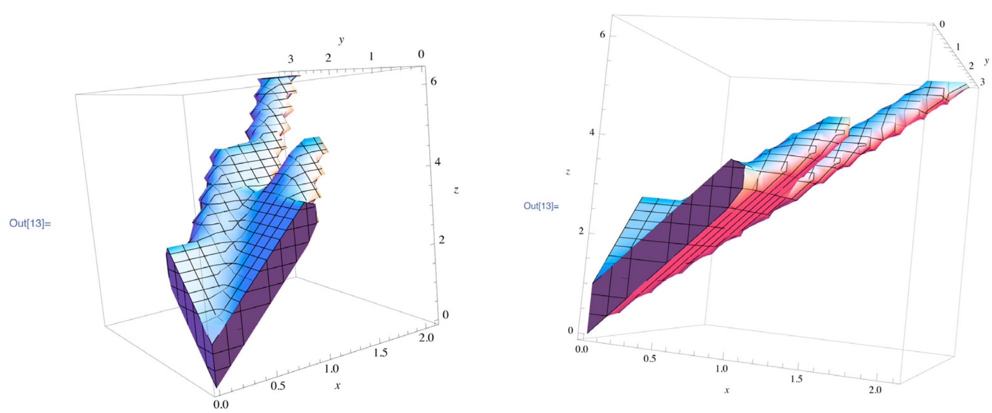

Fig. 1. Hilbert–Kunz multiplicity: view from front and side. $e_{HK}(\mathcal{B}_R(I, J)) = \frac{41}{18}$..

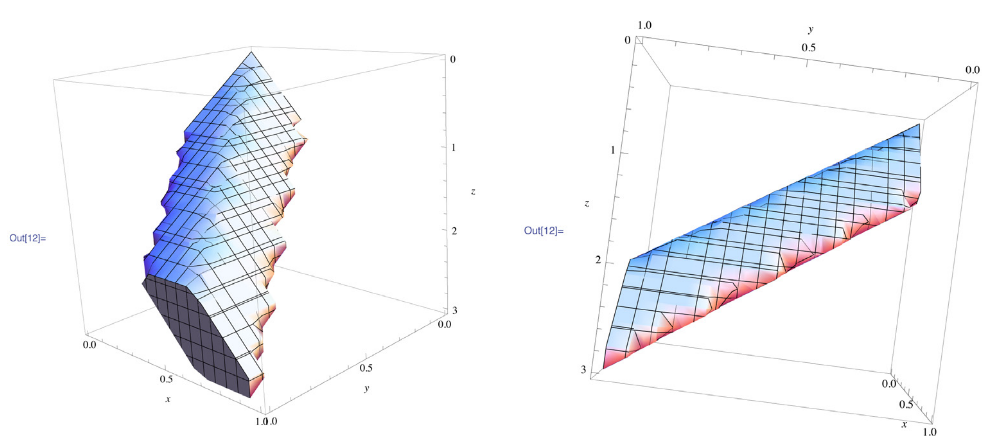

Fig. 2. F-Signature: view from front and side. $s(\mathcal{B}_R(I, J)) = \frac{11}{36}$..

Likewise, $e_{HK}(\mathcal{B}_R(J,J)) = \frac{22789}{12005}$ and $s(\mathcal{B}_R(J,J)) = \frac{1221}{12005}$ are easily obtained.

4. Impact

The significance of this software is due to the importance of the Hilbert–Kunz multiplicity and F-signature in the fields of algebraic geometry and commutative algebra. Countless articles have been published on the invariants, by myriad authors, too numerous to list. The overarching idea is that the Hilbert–Kunz multiplicity is a measure of the pathology of the singularities of the ring: if R is a regular ring, e.g., $R = \mathsf{k}[x_1, \ldots, x_n]$, then the Hilbert–Kunz multiplicity equals 1, and in general, the closer the invariant is to 1, the better the singularities of the ring. (Likewise, the F-signature of a regular ring is also 1.) For this reason, and because it is so difficult to compute, the Hilbert–Kunz multiplicity has been analyzed extensively in terms of upper and lower bounds. Our program now provides a large class of examples where the exact value is obtained, impacting the practice, and improving the resources, of those in the fields of research; it often executes in a matter of seconds or minutes a process that would take an individual an hour or more per example. Moreover, it provides the values of both the Hilbert–Kunz multiplicity and F-signature invariants. Results in the substantial literature exist which seek to relate the two invariants. For example, if $a_i = b_i$ for all i, then $e_{HK}(\mathcal{B}) + s(\mathcal{B}) = 2$ (see Example 3.2), [2, Proposition 4.3]. With the capability of our software, other relationships and properties may now be discovered and tested, and new research questions may be pursued.

The ability to calculate these invariants will, in turn, further the study of the structure of rings by perhaps answering questions on tight closure. The relationship of the Hilbert–Kunz multiplicity to the theory of tight closure is analogous to that of the Hilbert–Samuel multiplicity to integral closure. Moreover, tight closure ties into long-standing conjectures and questions in commutative algebra and algebraic geometry. (See, e.g., [1] and its references.)

Finally, it is important to note that our program works in any dimension. Consequently, light may be shed on questions and theories beyond small dimensions, which are often the limited proving grounds for results in the literature. Moreover, given the overlap of intersection algebras with objects in algebraic geometry, for one, the program's application and interest extends beyond a single field of study.

Acknowledgments

The authors thank Florian Enescu at Georgia State University, Dawn Wilkins and Conrad Cunningham in the Department of Computer Science at the University of Mississippi, and the various specialists in the Mississippi Center for Supercomputing Research on the Oxford campus for useful conversations about this project.

Conflict of interest

There is no conflict of interest.

References

[1] Huneke C. Hilbert-Kunz multiplicity and the F-Signature. In: Commutative algebra: Expository papers dedicated to David Eisenbud on the occasion of his 65th birthday. New York: Spring; 2013, p. 485–525.
[2] Enescu F, Spiroff S. Computing the invariants of intersection algebras of principal monomial ideals. Int J Alg Comp 2019;29(2).
[3] Eto K. Multiplicity and Hilbert-Kunz multiplicity of monoid rings. Tokyo J Math 2002;25(2):241–5.
[4] von Korff M. The F-signature of toric varieties (Ph.D. thesis), University of Michigan; 2012,
[5] Watanabe K, Yoshida K. Hilbert-Kunz multiplicity of three dimensional local rings. Nagoya Math J 2005;177:47–75.
[6] Watanabe K. Hilbert-Kunz multiplicity of toric rings. Inst Nat Sci Nihon Univ 2000;35:173–7.
[7] Malec S. On the intersection algebra of principal ideals. Comm Algebra 2012;43(2):623–35.
[8] Enescu F, Malec S. Intersection algebras for principal monomial ideals in polynomial rings. J Alg Appl 2015;14(7):8–23.
[9] Wolfram Research, Inc. Mathematica, versions 9 and above. Champaign, Illinois; 2014.
[10] Cortadellas Benítez T, D'Andrea C, Enescu F. On the resolution of fan algebras of principal ideals over noetherian rings. 2017; https://arxiv.org/abs/1612.02939.