

```

gap> g:= SymmetricGroup( 4 );
Sym( [ 1 .. 4 ] )
gap> tbl:= CharacterTable( g ); HasIrr( tbl );
false
0 1 2 3 4
gap> tblmod2:= CharacterTable( tbl, 2 );
o5 = total: 1 4 13 14 4
0: 1 . . . .
1: . 2 2 4 2
2: . 2 5 6 .
3: . . 4 . 2
4: . . . 4 .
5: . . 2 . .
gap> tblmod2 = BrauerTable( tbl, 2 );
o5 : BrauerTable
16 : betti(t,Weights=>{0,true}
0 1 2 3 4
gap> libtbl:= CharacterTable( "M" );
o6 = total: 1 4 13 14 4
0: 1 . . . .
1: . 2 2 4 2
2: . 2 5 6 .
3: . . 4 . 2
4: . . . 4 .
5: . . 2 . .
gap> libtbl:= CharacterTableRegular( libtbl, 2 );
o6 : BettiTally
17 : t1 = betti(t,Weights=>{1,1})
0 1 2 3 4
gap> ComputedBrauerTables( tbl );
[ , BrauerTable( Sym( [ 1 .. 4 ] ), 2 ), ]
o7 = total: 1 4 13 14 4
0: 1 . . . .
1: . . . . .
2: . . . . .
3: . 2 . . .
4: . . . . .
5: . 2 . . .
6: . . 1 . .
7: . . 8 6 .
8: . . 4 8 4
ring r1 = 32003,(x,y,z),ds;
int a,b,c,t=11,5,3,0;
poly f = x^a+y^b+z^(3*c)+x^(c+2)*y^(c-1)+x^(c-2)*y^c*(y^2+t*x)^2;
option(noprot);
timer=1;
ring r2 = 32003,(x,y,z),dp;
poly f=imap(r1,f);
ideal j=jacob(f);
vdim(std(j));
==> 536
vdim(std(j+f));
==> 195
timer=0; // reset timer
o8 = BettiTally{ (0, {0, 0}, 0) => 1 }
(1, {2, 2}, 4) => 2
(1, {3, 3}, 6) => 2
(2, {3, 7}, 10) => 2
(2, {4, 4}, 8) => 1
(2, {4, 5}, 9) => 4
(2, {5, 4}, 9) => 4
(2, {7, 4}, 11) => 4
(3, {4, 7}, 12) => 2
(3, {5, 5}, 10) => 6
(3, {7, 4}, 11) => 4
(4, {5, 7}, 12) => 2
(4, {7, 5}, 12) => 2

```

Journal of Software for Algebra and Geometry

Calculations involving symbolic powers

BEN DRABKIN, ELOÍSA GRIFO,
ALEXANDRA SECELEANU AND BRANDEN STONE

Calculations involving symbolic powers

BEN DRABKIN, ELOÍSA GRIFO,
ALEXANDRA SECELEANU AND BRANDEN STONE

ABSTRACT: Symbolic powers is a classical commutative algebra topic that relates to primary decomposition, consisting, in some circumstances, of the functions that vanish up to a certain order on a given variety. However, these are notoriously difficult to compute, and there are seemingly simple questions related to symbolic powers that remain open even over polynomial rings. In this paper, we describe a Macaulay2 software package that allows for computations of symbolic powers of ideals and which can be used to study the equality and containment problems, among others.

1. INTRODUCTION. Given an ideal I in a Noetherian domain R , the n -th *symbolic power* of I is the ideal defined by

$$I^{(n)} = \bigcap_{P \in \text{Ass}(I)} (I^n R_P \cap R). \quad (1-1)$$

When I has no embedded primes, the minimal primes of I^n coincide with the associated primes of I , and $I^{(n)}$ as above corresponds to the intersection of the primary components corresponding to minimal primes of I^n . In particular, under these circumstances the definition is unchanged if instead we have P ranging over the set of minimal associated primes $\text{Min}(I)$. However, if we consider any ideal I , with no assumptions on its associated primes, there are two possible notions of symbolic powers: the one above and the one given by

$$I^{(n)} = \bigcap_{P \in \text{Min}(I)} (I^n R_P \cap R). \quad (1-2)$$

The `SymbolicPowers.m2` package allows the user to compute the symbolic powers of any ideal over a polynomial ring, using the definition of symbolic powers given in (1-1) as the standard, but allowing the user to take the definition in (1-2) instead via the option `UseMinimalPrimes`. This option can be used in any method included in the package.

MSC2010: primary 13P99; secondary 13A15, 13C99.

Keywords: symbolic powers, Macaulay2.

`SymbolicPowers.m2` version 2.0

Symbolic powers are a classical topic that relates to many subjects within commutative algebra and algebraic geometry, and is an active area of current research. If P is a prime ideal in a regular ring, the classical Zariski–Nagata theorem [Zariski 1949; Nagata 1962] says that the symbolic powers of P consist of the functions that vanish up to order n in the corresponding variety. For a polynomial ring over a perfect field, these coincide with differential powers. For a survey on symbolic powers, see [Dao et al. 2018].

Various invariants have been defined to compare symbolic and ordinary powers of ideals: the resurgence [Bocci and Harbourne 2010], the Waldschmidt constant [Bocci and Harbourne 2010], and the symbolic defect [Galetto et al. 2019], among others. Using the `SymbolicPowers.m2` package, these can be in some cases explicitly computed and in others approximated.

2. BASIC USAGE. The main method in the `SymbolicPowers.m2` package is `symbolicPower`, which takes as inputs an ideal I and an integer n and returns $I^{(n)}$. Computations are done using the standard definition of symbolic powers; if the option `UseMinimalPrimes` is set true, then the definition of symbolic powers used in the computations will be the nonstandard one, as described in the introduction. When `UseMinimalPrimes` is set true, the algorithm takes a primary decomposition of I^n and intersects the components corresponding to minimal primes. Throughout the rest of the paper, we will assume that the `UseMinimalPrimes` option is set to false, which is the default setting.

Various algorithms are used for the computation of symbolic powers. This package follows the order given below to decide the optimal algorithm applicable for computing `symbolicPower(I,n)`:

- (1) If I is a squarefree monomial ideal, the routine intersects the n -th powers of the associated primes of I .
- (2) If I is a monomial ideal, but not squarefree, the routine takes a primary decomposition of I and intersects the n -th powers of the intersections of the primary components associated to primes contained in each maximal element of $\text{Ass}(I)$ (see [Cooper et al. 2017, Lemma 3.1]).
- (3) If I is a saturated homogeneous ideal whose height is one less than the dimension of its ambient ring, the routine returns the saturation of I^n with respect to the maximal ideal.
- (4) If I is height unmixed (meaning that all the associated primes of I have the same height) the routine computes the top dimensional components of I^n using an algorithm of Eisenbud, Huneke and Vasconcelos [Eisenbud et al. 1992] (see Section 3).

- (5) If all else fails, the routine compares the radicals of a primary decomposition of I^n with the associated primes of I , and intersects the components corresponding to minimal primes.

Whenever primary decomposition is computed, the package uses the existing Macaulay2 routine for computing primary decompositions, which by default employs the Shimoyama–Yokoyama algorithm [1996] except when the given ideal is monomial. However, note that finding primary decompositions is generally a fairly slow process, and certainly slower than the first four strategies listed above. Explicit experiments demonstrating that the first, third and fourth strategies outperform the last, even when factoring in the time needed to check their applicability, are given in Examples 2.1, 2.2 and 2.4. For this reason, we avoid computing the primary decomposition of I^n whenever possible.

There is one notable exception to this philosophy: in the case when the primary components of an ideal are complete intersections, the extra time spent computing a primary decomposition can be worth it (cf. Example 2.5). If the option `CIPrimes` is set to `true`, then `symbolicPower(I,n)` outputs the intersection of the n -th powers of the primary components of the input ideal I , if each of these components is a complete intersection and they all have the same height. Using the `CIPrimes` option computes the symbolic power much more quickly than the other five strategies in cases when there are sufficiently many associated primes.

We compare below the running times of the various algorithms that we use for computing symbolic powers in several examples. In the following, we denote the first algorithm listed above by `mon'1`, the third by `sat`, the fourth by `unmixed`, and the last by `pdec`.

Example 2.1. Set $R = k[x, y, z]$, where k is a field of characteristic not equal to 2, and

$$I = (x(y^3 - z^3), y(z^3 - x^3), z(x^3 - y^3))$$

is an ideal which has become known in the literature as a Fermat ideal. The table below compares the running times in seconds for the algorithms `pdec` and `sat` as well as the total running time for `symbolicPower(I,5)`. Note that in this example the `symbolicPower` method checks the hypotheses needed for applying the saturation algorithms and then runs this routine:

	<code>pdec</code>	<code>sat</code>	<code>symbolicPower</code>
running times for $I^{(5)}$	4	0.036	0.040

Example 2.2. Set $R = k[x_1, x_2, x_3, x_4, x_5]$ and let I be the ideal generated by all the squarefree monomials of degree 2 in R . The running times in seconds for the algorithms `pdec` and `mon'1` are compared to the running time for `symbolicPower(I,5)`

in the following table:

	pdec	mon'1	symbolicPower
running times for $I^{(5)}$	1.35	0.004	0.004

Example 2.3. Set $R = k[x, y, z]$ and let $I = (xy, xz, yz)$. In this example we compare the mon'1 and sat strategies, since both are applicable. The running times in seconds for the algorithms pdec, sat and mon'1 are compared to the running time for symbolicPower, which also checks the applicability of the mon'1 strategy.

	mon'1	sat	pdec	symbolicPower
running times for $I^{(5)}$	0.001	0.006	0.021	0.002
running times for $I^{(10)}$	0.001	0.369	0.558	0.002

Example 2.4. Set $R = k[x_1, \dots, x_{12}]$ and let I be the ideal generated by the 2×2 minors of a generic 3×4 matrix with entries the variables of R . The running times in seconds for the algorithms unmixed and pdec are compared to the running time for symbolicPower(I, 5) in the following table:

	unmixed	pdec	symbolicPower
running times for $I^{(5)}$	3.970	44.538	4.231

This example shows that even including the overhead of checking that the ideal above is height unmixed, the routine symbolicPower, which in this case uses the unmixed strategy based on the method of Eisenbud, Huneke and Vasconcelos, outperforms the pdec algorithms.

Example 2.5. Let I be the ideal of ten general points in $\mathbb{P}^2$. We illustrate the computation times for the fifth symbolic powers of I with the option CIPrimes turned on in comparison to the default strategy for this case, which is to use the saturation algorithm.

	CIPrimes	sat	symbolicPower
running times for $I^{(5)}$	0.447	3.483	3.495

3. APPLICATIONS.

Methods based on a result of Eisenbud, Huneke, and Vasconcelos. We can identify the heights of all the associated primes of an ideal in a regular ring using the following result:

Theorem 3.1 [Eisenbud et al. 1992]. *Given an ideal I in a regular domain R of height h , then for each $e \geq h$, I has an associated prime of height e if and only if the height of $\text{Ext}^e(R/I, R)$ is e . In particular, the intersection of the top dimensional components of I is given by $\text{Ann Ext}_R^h(R/I, R)$.*

The already existing method `topComponents`, also based on this result, returns the intersection of the primary components of minimal height of an ideal. In particular, if I has pure height h , then `topComponents( $I^n$ )` returns $I^{(n)}$. This is one of the strategies used by the method `symbolicPower`.

Further, the `SymbolicPowers.m2` package also includes the method `bigHeight`, which computes the largest height of an associated prime of I , and the method `assPrimesHeight`, which returns a list of all the heights of the associated primes of I . Both of these are based on Theorem 3.1.

The method `minimalPart` returns the intersection of the minimal components of a given ideal, which is in general different from `topComponents`. Instead of explicitly finding the associated primes of I and taking their heights, Theorem 3.1 is used.

Equality. Symbolic powers do not, in general, coincide with the ordinary powers, even in the case of prime ideals. In fact, the question of characterizing the ideals I for which $I^{(n)} = I^n$ for all n is essentially open. One can determine whether the n -th symbolic and ordinary powers of a given ideal coincide using `isSymbolicEqualOrdinary`, often without computing the actual symbolic power of I . For this, the package makes use of `bigHeight`. To determine whether $I^{(n)} = I^n$ for a specific value of n , `isSymbolicEqualOrdinary` first compares the big heights of I^n and I : if the big heights differ, then I^n must have embedded components, and `isSymbolicEqualOrdinary` returns `false`; if the big heights are both equal to the height of I , then I^n cannot have embedded components, and `isSymbolicEqualOrdinary` returns `true`. This is faster than computing the set of associated primes of I^n . Using `symbolicDefect`, one can quantify the difference between I^m and $I^{(m)}$ by computing the symbolic defect of I in the power m , defined by Galetto, Geramita, Shin, and Van Tuyl in [Galetto et al. 2019] to be the minimal number of generators of $I^{(m)}/I^m$.

The packing problem. Besides allowing the user to determine when $I^{(n)} = I^n$ holds without the need to explicitly compute $I^{(n)}$, the `SymbolicPowers.m2` package also includes other methods that can be applied to this question. In particular, the package includes methods related to the packing problem, which was originally formulated in the context of max-flow min-cut properties by Conforti and Cornuéjols [1990]. Work of Gitler, Villarreal and others shows that this problem can be rewritten as a conjectural characterization of the squarefree monomial ideals having $I^{(n)} = I^n$ for all n as those ideals that satisfy the packing property. The

method `isPacked` determines if a given squarefree monomial ideal has this property. In particular, should the packing problem have an affirmative answer, this method could be used as a test for whether the equality $I^{(n)} = I^n$ holds for all n . For a quick survey on the packing problem, see [Dao et al. 2018].

The containment problem. The containment problem for ordinary and symbolic powers of ideals consists of answering the following question: given an ideal I , for which values of a and b does the containment $I^{(a)} \subseteq I^b$ hold? Over a regular ring, a well known theorem of Ein, Lazarsfeld and Smith [2001], Hochster and Huneke [2002], and Ma and Schwede [2018] gives a partial answer to that question: when I is a radical ideal, $I^{(hn)} \subseteq I^n$ holds for all n , where h denotes the big height of the ideal I . However, this is not necessarily best possible; see [Szemberg and Szpond 2017] for a survey. Using `containmentProblem`, the user can determine the smallest value of a , given b , for which $I^{(a)} \subseteq I^b$. Conversely, using the option `InSymbolic`, the user can determine the largest value of b , given a , for which $I^{(a)} \subseteq I^b$.

Example 3.2 (containment problem).

```
i1 : loadPackage "SymbolicPowers";
i2 : R=QQ[x,y,z];
i3 : I=ideal(x*(y^3-z^3),y*(z^3-x^3),z*(x^3-y^3));
o3 : ideal of R
i4 : containmentProblem(I,2)
o4 : 4
i6 : containmentProblem(I,5, InSymbolic=>true)
o6 : 3
```

The computation `containmentProblem(I,2)=4` illustrated above should be interpreted as stating that $I^{(4)} \subseteq I^2$ and $I^{(3)} \not\subseteq I^2$, while we can interpret the computation `containmentProblem(I,5,InSymbolic=>true)=3` as stating that $I^{(5)} \subseteq I^3$ and $I^{(5)} \not\subseteq I^4$.

Other applications. Some of the other methods in the package include specialized functionality for computations in positive characteristic and for computations specific to ideals defining monomial curves.

The method `symbolicPowerPrimePosChar` gives another algorithm for computing symbolic powers which is specific to working in prime characteristic p . This method can be faster than the other algorithms for computing symbolic powers $I^{(n)}$ for values of n very close to being a power of p , but not for general values of n .

For the special case of monomial curves $k[t^{a_1}, \dots, t^{a_k}]$, both of the methods `symbolicPowerMonomialCurve` and `containmentProblemMonomialCurve` essentially run `symbolicPower` and `containmentProblem`.

4. ASYMPTOTIC INVARIANTS. In an effort to make progress on the containment problem, various asymptotic interpolation invariants have been proposed by Bocci and Harbourne [2010]. One such invariant is the Waldschmidt constant for a homogeneous ideal I . This is an asymptotic measure of the initial degree of the symbolic powers of I . The *initial degree* of a homogeneous ideal I is $\alpha(I) = \min\{d \mid I_d \neq 0\}$, i.e., the smallest degree of a nonzero element in I . The *Waldschmidt constant* of I is defined to be

$$\widehat{\alpha}(I) = \lim_{m \rightarrow \infty} \frac{\alpha(I^{(m)})}{m}.$$

Due to the asymptotic nature of the Waldschmidt constant, there is no a priori algorithm to determine this invariant for arbitrary ideals, although the initial degrees of individual symbolic powers can be computed using `minDegreeSymbPower`. An important exception is the case when the ideal I is a monomial ideal. In this context, the Waldschmidt constant can be computed as the smallest among the sums of the coordinates of all points in a convex body termed the *symbolic polyhedron* of I [Cooper et al. 2017; Bocci et al. 2016]. Our package computes Waldschmidt constants of monomial ideals by finding their symbolic polyhedron. The `symbolicPolyhedron` routine makes heavy use of the `Polyhedra.m2` package by René Birkner, which in turn relies on the `FourierMotzkin.m2` package by Greg Smith. This allows to determine the Waldschmidt constants of monomial ideals exactly as in the following example.

Example 4.1 (Waldschmidt constant of monomial ideals).

```
i1 : loadPackage "SymbolicPowers";
i2 : R=QQ[x,y,z];
i3 : I=ideal(x*y,x*z,y*z);
i4 : symbolicPolyhedron(I)
o4 = {ambient dimension => 3
      dimension of lineality space => 0
      dimension of polyhedron => 3
      number of facets => 6
      number of rays => 3
      number of vertices => 4
o4 : Polyhedron
i5 : waldschmidt I
Ideal is monomial, the Waldschmidt constant is computed exactly
3
o5 = -
2
o5 : QQ
```

In the case of arbitrary ideals, the Waldschmidt constant is approximated by

taking the minimum of the values $\alpha(I^{(m)})/m$, where m ranges from 1 to a specified optional input `SampleSize`.

Example 4.2 (Waldschmidt constant of arbitrary ideals).

```
i1 : loadPackage "SymbolicPowers";
i2 : R=QQ[x,y,z];
i3 : I=ideal(x*(y^3-z^3),y*(z^3-x^3),z*(x^3-y^3));
o3 : Ideal of R
i4 : waldschmidt I
Ideal is not monomial, the Waldschmidt constant is approximated
using first 5 powers.
o4 = 3
o4 : QQ
```

Note that the true value for the Waldschmidt constant of the above ideal is indeed 3 as proven in [Dumnicki et al. 2015]. In general, for an ideal that is not monomial, the function `waldschmidt` will return an upper bound on the true value of the Waldschmidt constant.

Another asymptotic invariant termed *resurgence* [Bocci and Harbourne 2010] is defined as

$$\rho(I) = \sup \left\{ \frac{m}{r} \mid I^{(m)} \not\subseteq I^r \right\}.$$

There are no algorithms known to date that compute resurgence exactly; therefore, our package computes a lower bound for the resurgence by taking the maximum of the values $\frac{m}{r}$, where r ranges from 1 to the optional input `SampleSize`.

Continuing with the ideal in the previous example, we compute a lower bound on its resurgence using the default `SampleSize`, which is 5, and also a custom `SampleSize`. As expected, the lower bound increases as the `SampleSize` is increased, i.e., a larger `SampleSize` produces a better lower bound.

Example 4.3 (lower bound on resurgence).

```
i1 : loadPackage "SymbolicPowers";
i2 : R=QQ[x,y,z];
i3 : I=ideal(x*y,x*z,y*z);
i5 : lowerBoundResurgence(I)
6
o5 = -
5
o5 : QQ
i6 : lowerBoundResurgence(I,SampleSize=>10)
5
o6 = -
4
o6 : QQ
```

ACKNOWLEDGMENTS. We would like to thank the organizers of the July 2017 Macaulay2 Workshop at the University of California, Berkeley, where a large portion of this work was done. The code for computing the symbolic polyhedron and Waldschmidt constant of a monomial ideal was developed by Seceleanu in collaboration with Andrew Conner and Xuehua (Diana) Zhong. We thank them for their contribution to these routines.

We also thank the anonymous referee for many helpful suggestions and comments.

This research was partially supported by NSF grant DMS #1502282; we thank Luis Núñez Betancourt and Craig Huneke for that support, and for organizing a conference at the University of Virginia where part of this work took place. Seceleanu was supported by NSF grant DMS#1601024. Drabkin and Seceleanu were supported by EPSCoR grant OIA–1557417.

SUPPLEMENT. Version 2.0 of `SymbolicPowers.m2` is contained in the [online supplement](#).

REFERENCES.

- [Bocci and Harbourne 2010] C. Bocci and B. Harbourne, “Comparing powers and symbolic powers of ideals”, *J. Algebraic Geom.* **19**:3 (2010), 399–417. [MR](#) [Zbl](#)
- [Bocci et al. 2016] C. Bocci, S. Cooper, E. Guardo, B. Harbourne, M. Janssen, U. Nagel, A. Seceleanu, A. Van Tuyl, and T. Vu, “The Waldschmidt constant for squarefree monomial ideals”, *J. Algebraic Combin.* **44**:4 (2016), 875–904. [MR](#) [Zbl](#)
- [Conforti and Cornuéjols 1990] M. Conforti and G. Cornuéjols, “A decomposition problem for balanced matrices”, working paper 1990–10, Carnegie Mellon Univ., Tepper School of Business, 1990.
- [Cooper et al. 2017] S. M. Cooper, R. J. D. Embree, H. T. Hà, and A. H. Hoefel, “Symbolic powers of monomial ideals”, *Proc. Edinb. Math. Soc.* (2) **60**:1 (2017), 39–55. [MR](#) [Zbl](#)
- [Dao et al. 2018] H. Dao, A. De Stefani, E. Grifo, C. Huneke, and L. Núñez Betancourt, “Symbolic powers of ideals”, pp. 387–432 in *Singularities and foliations: geometry, topology and applications* (Salvador, Brazil, 2015), edited by R. N. Araújo dos Santos et al., Springer Proc. Math. Stat. **222**, Springer, 2018. [MR](#) [Zbl](#)
- [Dumnicki et al. 2015] M. Dumnicki, B. Harbourne, U. Nagel, A. Seceleanu, T. Szemberg, and H. Tutaj-Gasińska, “Resurgences for ideals of special point configurations in $\mathbb{P}^N$ coming from hyperplane arrangements”, *J. Algebra* **443** (2015), 383–394. [MR](#) [Zbl](#)
- [Ein et al. 2001] L. Ein, R. Lazarsfeld, and K. E. Smith, “Uniform bounds and symbolic powers on smooth varieties”, *Invent. Math.* **144**:2 (2001), 241–252. [MR](#) [Zbl](#)
- [Eisenbud et al. 1992] D. Eisenbud, C. Huneke, and W. Vasconcelos, “Direct methods for primary decomposition”, *Invent. Math.* **110**:1 (1992), 207–235. [MR](#) [Zbl](#)
- [Galetto et al. 2019] F. Galetto, A. V. Geramita, Y.-S. Shin, and A. Van Tuyl, “The symbolic defect of an ideal”, *J. Pure Appl. Algebra* **223**:6 (2019), 2709–2731. [MR](#) [Zbl](#)
- [Hochster and Huneke 2002] M. Hochster and C. Huneke, “Comparison of symbolic and ordinary powers of ideals”, *Invent. Math.* **147**:2 (2002), 349–369. [MR](#) [Zbl](#)
- [Ma and Schwede 2018] L. Ma and K. Schwede, “Perfectoid multiplier/test ideals in regular rings and bounds on symbolic powers”, *Invent. Math.* **214**:2 (2018), 913–955. [MR](#) [Zbl](#)

- [Nagata 1962] M. Nagata, *Local rings*, Interscience Tracts in Pure Appl. Math **13**, Interscience, New York, 1962. [MR](#) [Zbl](#)
- [Shimoyama and Yokoyama 1996] T. Shimoyama and K. Yokoyama, “Localization and primary decomposition of polynomial ideals”, *J. Symbolic Comput.* **22**:3 (1996), 247–277. [MR](#) [Zbl](#)
- [Szemberg and Szpond 2017] T. Szemberg and J. Szpond, “On the containment problem”, *Rend. Circ. Mat. Palermo* (2) **66**:2 (2017), 233–245. [MR](#) [Zbl](#)
- [Zariski 1949] O. Zariski, “A fundamental lemma from the theory of holomorphic functions on an algebraic variety”, *Ann. Mat. Pura Appl.* (4) **29** (1949), 187–198. [MR](#) [Zbl](#)

RECEIVED: 5 Dec 2017

REVISED: 2 Feb 2019

ACCEPTED: 20 May 2019

BEN DRABKIN:

benjamin.drabkin@huskers.unl.edu

Department of Mathematics, University of Nebraska, Lincoln, NE, United States

ELOÍSA GRIFO:

grifo@umich.edu

Department of Mathematics, University of Michigan, Ann Arbor, MI, United States

ALEXANDRA SECELEANU:

aseceleanu@unl.edu

Department of Mathematics, University of Nebraska, Lincoln, NE, United States

BRANDEN STONE:

bstone@hamilton.edu

Mathematics Department, Hamilton College, Clinton, NY, United States

<i>Strongly stable ideals and Hilbert polynomials</i>	1
Davide Alberelli and Paolo Lella	
<i>DiffAlg: a Differential algebra package</i>	11
Manuel Dubinsky, César Massri, Ariel Molinuevo and Federico Quallbrunn	
<i>Matroids: a Macaulay2 package</i>	19
Justin Chen	
<i>Computing quasidegrees of A-graded modules</i>	29
Roberto Barrera	
<i>An algorithm for enumerating difference sets</i>	35
Dylan Peifer	
<i>Hyperplane arrangements in CoCoA</i>	43
Elisa Palezzato and Michele Torielli	
<i>Numerical implicitization</i>	55
Justin Chen and Joe Kileel	
<i>Random Monomial Ideals: a Macaulay2 package</i>	65
Sonja Petrović, Despina Stasi and Dane Wilburne	
<i>Calculations involving symbolic powers</i>	71
Ben Drabkin, Eloísa Grifo, Alexandra Seceleanu and Branden Stone	
<i>The gfanlib interface in Singular and its applications</i>	81
Anders Jensen, Yue Ren and Hans Schönemann	