Journal of Software for Algebra and Geometry

The SpaceCurves package in Macaulay2

Mengyuan Zhang

vol 8 2018

JSAG **8** (2018), 31–48
dx.doi.org/10.2140/jsag.2018.8.31

The Journal of Software for
Algebra and Geometry

The SpaceCurves package in Macaulay2

MENGYUAN ZHANG

ABSTRACT: This note introduces the Macaulay2 package `SpaceCurves.m2` with illustration. The 1.0 version of the package, provided in the accompanying online supplement, is devoted to the generation of three types of curves in $\mathbb{P}^3$: smooth curves, ACM curves and curves that are minimal in the even liaison class.

1. INTRODUCTION. The `SpaceCurves` project was initiated by R. Hartshorne, F. Schreyer and M. Stillman during the 2017 Macaulay2 workshop at UC Berkeley. Since the workshop, Zhang has improved old code and developed the package into the present version. The goal of the `SpaceCurves` package is to generate three types of curves in $\mathbb{P}^3$; smooth curves, ACM curves, and minimal curves in a given even liaison class.

In Section 2 we illustrate how to produce smooth curves exhausting all possibilities of (degree, genus) pairs. The philosophy is the following: first we construct three types of surfaces; the smooth quadric surface, smooth cubic surfaces and rational quartic surfaces with a double line. Next, we construct divisors on these surfaces. Finally, we generate a random curve in a given divisor class.

In Section 3 we illustrate the stratification of the Hilbert scheme of ACM curves in $\mathbb{P}^3$ using Betti tables. We illustrate how to produce ACM curves exhausting all possibilities of Betti tables. First, we list all the possible Hilbert functions of ACM curves of a given degree, then we produce all Betti tables of ACM curves with a given Hilbert function. Finally, we generate a matrix of random forms with degrees specified by the Hilbert–Burch degree matrix and take the ideal of maximal minors.

In Section 4 we explain the construction of a curve that is minimal in its even liaison class from a given finite length module. The even liaison class can be specified by either the ideal of a curve in the even liaison class, or by a finite length module called the Hartshorne–Rao module. The implementation of the minimal curve algorithm follows the paper by Guarrera et al. [1997].

To make our computations exact, and to avoid coefficient explosion, we work over large finite fields $\mathbb{Z}/p\mathbb{Z}$ in Macaulay2. Over a small prime field, such as $\mathbb{Z}/2\mathbb{Z}$

MSC2010: 14-04, 14H50, 14HXX.
Keywords: space curves, minimal curves, Rao module, liaison theory, ACM curves, smooth curves.
SpaceCurves version 1.0

or $\mathbb{Z}/3\mathbb{Z}$, many of the curves produced will be singular. In these cases we refer the readers to the package `RandomCurvesOverVerySmallFiniteFields.m2` by C. Bopp and F. Schreyer.

2. Smooth curves in projective three-space.

For which pairs of integers (d, g) does there exist a connected smooth curve in $\mathbb{P}^3$ of degree d and genus g? G. Halphen gave partial answers in his prize winning treatise in 1882, and the complete solution to this question was given by Gruson and Peskine almost a century later in characteristic 0 and extended to characteristic p by Hartshorne.

(1) There are smooth plane curves of genus $g = \frac{1}{2}(d-1)(d-2)$ for any $d \geq 1$.

(2) (Castelnuovo) If a smooth curve does not lie on any plane, we must have

$$g \leq \left\lfloor \tfrac{1}{4}d^2 - d + 1 \right\rfloor.$$

Any curve obtaining this bound lies on a quadric surface.

(3) For each $a, b > 0$, there are smooth curves on the smooth quadric surface with degree $d = a + b$ and genus $g = (a-1)(b-1)$.

(4) On the quadric cone, if $d = 2a$ is even, there are smooth complete intersections of the quadric cone with another surface of degree a. If $d = 2a+1$ is odd, then any degree d curve on the quadric cone has genus $g = a^2 - a$.

(5) [Halphen 1882] If a curve does not lie on any plane or quadric surface, then

$$g \leq \tfrac{1}{6}d(d-3) + 1.$$

(6) [Gruson and Peskine 1982, Corollary 2.3] For $d \geq 1$ and

$$\tfrac{1}{\sqrt{3}}d^{3/2} - d + 1 < g \leq \tfrac{1}{6}d(d-3) + 1,$$

there is a smooth curve with degree d and genus g on a smooth cubic surface.

(7) [Gruson and Peskine 1982, Theorem 1.1; Hartshorne 1982, Theorem 0.2] For $d \geq 1$ and

$$0 \leq g \leq \tfrac{1}{8}(d-1)^2,$$

there is a smooth curve with degree d and genus g on a smooth quartic surface with a double line.

The `SpaceCurves` package generates curves on the surfaces mentioned above. Let us start from curves on the smooth quadric surface.

2.1. *Curves on a smooth quadric surface.*

To create a smooth quadric surface Q, we use the method function `quadricSurface(Ring)` where the input ring is taken as the ambient coordinate ring of $\mathbb{P}^3$. The output is a type of hashtable called `QuadricSurface`. It contains the following information:

```
i1 : needsPackage "SpaceCurves";
i2 : R = ZZ/101[x_0..x_3];
i3 : Q = quadricSurface(R)
o3 = ideal(- x x  + x x )
              1 2    0 3
o3 : QuadricSurface
i4 : peek Q
o4 = QuadricSurface{CanonicalClass => {-2, -2}    }
                    HyperplaneClass => {1, 1}
                    Ideal => ideal(- x x  + x x )
                                       1 2    0 3
                    IntersectionPairing => | 0 1 |
                                           | 1 0 |
```

Since $Q \cong \mathbb{P}^1 \times \mathbb{P}^1$, a basis of Pic$(Q)$ consists of $\pi_1^*(\mathscr{O}_{\mathbb{P}^1}(1))$ and $\pi_2^*(\mathscr{O}_{\mathbb{P}^1}(1))$, where π_1 and π_2 are the two canonical projections to $\mathbb{P}^1$. In this basis, the intersection pairing matrix is $\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]$ and the canonical class has coordinates $\{-2, -2\}$. The divisor class $\{1, 1\}$ in the given basis defines the embedding of Q in $\mathbb{P}^3$.

The method function `divisor(List,QuadricSurface)` produces divisors on the quadric surface Q. The output is a type of hashtable called `Divisor`, which carries a list that encodes the coordinates of the divisor as well as the surface it is on. The intersection number is computed from the coordinates of D and E using the intersection matrix of Q.

```
i5 : D = divisor({2,3},Q)
o5 = {2, 3}
o5 : Divisor
i6 : peek D
o6 = Divisor{Coordinate => {2, 3}         }
             Surface => ideal(- y*z + x*w)
i7 : E = divisor({1,2},Q);
i8 : D*E
o8 = 7
```

Since we can compute the intersection number of any two divisors on Q, we can compute the degree and arithmetic genus of a divisor using Bezout's theorem and the adjunction formula:

$$\deg D = D.H,$$

$$2p_a(D) - 2 = D.(D+K),$$

where H and K denote the hyperplane class and the canonical class, respectively. The advantage of computing the degree and genus of a divisor abstractly is that the coordinates of the divisor can be in any ring that supports rational arithmetic. In particular, we can verify the formula for the degree and genus of a divisor of type (a, b) on Q.

```
i9 : S = QQ[a,b];
i10 : F = divisor({a,b},Q);
i11 : degree F
o11 = a + b
o11 : S
i12 : genus F
o12 = a*b - a - b + 1
o12 : S
```

We use the method function `curve(Divisor)` to produce a random curve in a given divisor class. The output type is called `Curve`, which is a hashtable that encodes the ideal of the curve as well as the `Divisor` it comes from. To extract the ideal, one could use the method `ideal(Curve)` or equivalently the key `Ideal` of the hashtable `Curve`.

```
i13 : C = curve D;
i14 : I = ideal C;
i15 : (degree I, degree D, genus I, genus D)
o15 = (5, 5, 2, 2)
o15 : Sequence
```

Here is how the method `curve(Divisor)` works for divisors on a smooth quadric surface. If D has coordinates $\{a, b\}$, we generate a random form f of bidegree (a, b) in the Cox ring

$$\mathrm{Cox} = k[s, t] \otimes_k k[u, v]$$

and create the Segre map $\psi : R = k[x, y, z, w] \to \mathrm{Cox}\,/(f)$ where

$$x \mapsto s \otimes u, \qquad y \mapsto s \otimes v, \qquad z \mapsto t \otimes u, \qquad w \mapsto t \otimes v.$$

The kernel of the map ψ will be the ideal of a curve in the divisor class (a, b) on the quadric surface. Since we are using random bihomogeneous forms, the output is typically smooth.

2.2. ***Curves on a smooth cubic surface.*** Although it is easy to produce a smooth cubic surface, it is not straightforward to generate curves of a given degree and genus on the given cubic surface. Instead, we realize the cubic surface as a blowup $\pi : X \to \mathbb{P}^2$ at six general points, anticanonically embedded into $\mathbb{P}^3$. Since the automorphisms of $\mathbb{P}^2$ act transitively on four distinct points, we fix four points,

$$P_1 = [1:0:0], \quad P_2 = [0:1:0], \quad P_3 = [0:0:1], \quad P_4 = [1:1:1],$$

and choose points P_5 and P_6 randomly. Then $\mathrm{Pic}(X) \cong \mathbb{Z}^7$ with a basis given by $L, -E_1, \ldots, -E_6$, where $L = \pi^*(\mathscr{O}_{\mathbb{P}^2}(1))$ and E_i is the exceptional divisor of the point P_i. The intersection matrix of X in this basis is given by

$$\begin{bmatrix} 1 & 0 \\ 0 & -\mathrm{Id} \end{bmatrix},$$

where Id is the 6×6 identity matrix. The complete linear system $|3L - E_1 - \cdots - E_6|$ is very ample and embeds X into $\mathbb{P}^3$ as a smooth cubic surface. The canonical class of X is $-3L + E_1 + \cdots + E_6$. See [Hartshorne 1977, V.4] for a treatment of these well known facts.

We can use the method function `cubicSurface(Ring)` to generate a smooth cubic surface. The output is a type of hashtable called `CubicSurface`. The key `BlowUpPoints` stores the list of ideals of the six points. The key `MapToP3` stores the rational map from $\mathbb{P}^2$ to $\mathbb{P}^3$ which comes from the restriction of the embedding $X \hookrightarrow \mathbb{P}^3$.

```
i5 : X = cubicSurface(R)
o5 = ideal(...)
o5 : CubicSurface
i6 : peek X
o6 = CubicSurface{BlowUpPoints => {...}                              }
                  CanonicalClass => {-3, -1, -1, -1, -1, -1, -1}
                  HyperplaneClass => {3, 1, 1, 1, 1, 1, 1}
                  IntersectionPairing => | 1 0  0  0  0  0  0  |
                                         | 0 -1 0  0  0  0  0  |
                                         | 0 0  -1 0  0  0  0  |
                                         | 0 0  0  -1 0  0  0  |
                                         | 0 0  0  0  -1 0  0  |
                                         | 0 0  0  0  0  -1 0  |
                                         | 0 0  0  0  0  0  -1 |
                  Ideal => ideal(...)
                  MapToP3 => ...
```

The equation of X in $\mathbb{P}^3$ is computed in the following way: let I be the ideal of the union of the six points $P_1, \ldots, P_6$. Since P_5 and P_6 are chosen randomly, we can presume that no three of P_i are collinear and that the points P_i do not all lie on any quadric. The Hilbert–Burch theorem thus determines the shape of the free resolution of I as

$$0 \to R(-4)^3 \xrightarrow{\phi} R(-3)^4 \to I \to 0,$$

where $R = k[y_0, y_1, y_2]$ is the coordinate ring of $\mathbb{P}^2$. Now ϕ is a 4×3 matrix of linear forms in three variables, which can be thought of as a $4 \times 3 \times 3$ tensor. There is a canonical way to consider ϕ as a $3 \times 3 \times 4$ tensor, and think of it as a 3×3 matrix of linear forms in four variables denoted by M'. The determinant of M' gives the defining equation of X in $\mathbb{P}^3$.

We can create divisors on X using `divisor(List,CubicSurface)`. Let us compute the degree and genus of a divisor on the smooth cubic surface.

```
i17 : S = QQ[a,b_1..b_6];
i18 : D = divisor(gens S,X);
i19 : degree D
o19 = 3a - b   - b   - b   - b   - b   - b
            1     2     3     4     5     6
i20 : genus D
o20 = ...
```

2.3. ***Curves on a rational quartic surface.*** Let us recall the construction of the quartic surface with a double line singularity from [Gruson and Peskine 1982]. Consider the blowup of $\mathbb{P}^2$ at nine general points $P_1, \dots, P_9$ given by $\pi : Y \to \mathbb{P}^2$. Let L denote the class $\pi^*(\mathscr{O}^1_{\mathbb{P}}(1))$ on Y, and let E_i denote the class of the exceptional divisor corresponding to the point P_i. Then $\operatorname{Pic}(Y)$ is isomorphic to the free abelian group generated by $L, -E_1, \dots, -E_9$. There is a unique smooth cubic curve Γ_0 on $\mathbb{P}^2$ passing through the nine points, and let Γ denote its proper transform on Y. We have $\Gamma = 3L - E_1 - \cdots - E_9$, and $-\Gamma$ is the canonical class of Y. If we set $C = L - E_1$, then the complete linear system $|C+\Gamma|$ is basepoint-free and maps Y into $\mathbb{P}^3$ as a quartic hypersurface. One can verify that the linear system $|C+\Gamma|$ separates points and tangent vectors for points in $Y - \Gamma$, and restricts to a degree 2 linear system on the elliptic curve Γ. As a result, $Y - \Gamma$ is mapped isomorphically onto its image, and Γ is mapped 2-1 to a line. The image of Y is a rational quartic surface with a double line singularity, where the double line is the scheme-theoretical image of the elliptic curve Γ. For proofs of these statements, see [Gruson and Peskine 1982, §1].

We use the method function `quarticSurfaceRational(Ring)` to create such a surface. We can verify that the singular locus of the image of Y is indeed the image of the elliptic Γ in Macaulay2.

```
i21 : Y = quarticSurfaceRational(R)
o21 = ideal(...)
o21 : QuarticSurfaceRational
i22 : peek Y
o22 = QuarticSurfaceRational{BlowUpPoints => {...
        CanonicalClass => {-3, -1, -1, -1, -1, -1, -1, -1, -1, -1}
        HyperplaneClass => {4, 2, 1, 1, 1, 1, 1, 1, 1, 1}
        IntersectionPairing => | 1 0  0  0  0  0  0  0  0  0  |
                               | 0 -1 0  0  0  0  0  0  0  0  |
                               | 0 0  -1 0  0  0  0  0  0  0  |
                               | 0 0  0  -1 0  0  0  0  0  0  |
                               | 0 0  0  0  -1 0  0  0  0  0  |
                               | 0 0  0  0  0  -1 0  0  0  0  |
                               | 0 0  0  0  0  0  -1 0  0  0  |
                               | 0 0  0  0  0  0  0  -1 0  0  |
                               | 0 0  0  0  0  0  0  0  -1 0  |
                               | 0 0  0  0  0  0  0  0  0  -1 |
          MapToP3 => ...
          Ideal => ...
i26 : pts = intersect(Y.BlowUpPoints);
i31 : S = ring pts;
i27 : hilbertFunction(3,module pts)
o27 = 1
i28 : G = gens trim pts;
i32 : gamma0 = ideal (G*random(source G,S^{-3}));
      -- This produces the unique plane cubic passing the 9 points.
o32 : Ideal of S
i36 : phi = map(S/gamma0,S);
i37 : psi = Y.MapToP3;
i39 : L = kernel (phi*psi)
o39 = ideal (y + 9918z - 2540w, x + 2540z - 11757w)
o39 : Ideal of R
      -- This is the image of Gamma
i44 : radical ideal jacobian Y.Ideal
o44 = ideal (y + 9918z - 2540w, x + 2540z - 11757w)
o44 : Ideal of R
      -- This is the singular locus of the image of Y
i45 : radical ideal jacobian Y.Ideal == L
o45 = true
```

The method function `divisor(List,QuarticSurfaceRational)` creates a divisor on the quartic surface with a double line. We explain how the method function `curve(Divisor)` turns `Divisors` into `Curves` on the `CubicSurface` and the `QuarticSurfaceRational`. Given a divisor $D = (a, b_1, \dots, b_n)$ where $n = 6$ or 9, we generate a random plane curve C_0 of degree a with multiplicity b_i at the point P_i and compute the equations of its image under the rational map

$$\mathbb{P}^2 \dashrightarrow \mathbb{P}^3.$$

Note that the produced curves have no components supported on the exceptional curves.

2.4. ***Generating smooth curves.*** We say a divisor class is smooth if its general members have smooth images in $\mathbb{P}^3$. The method function `smoothDivisors` generates all smooth divisors of a given degree on a given surface. So far the valid input surfaces are `QuadricSurface`, `CubicSurface` and `QuarticSurfaceRational`. On the smooth quadric surface as well as the smooth cubic surface, there are exact numerical criteria to decide whether D is smooth [Hartshorne 1977, Example V.4.8]. On the rational quartic surface however, we do not know of such a criterion, and thus our list is not exhaustive. One sufficient condition for D to be smooth is that D is basepoint-free and its restriction to Γ does not contain any pair of involution points as basepoints.

The following code generates smooth divisors of degree 4 on all the supported surfaces. Note that the output is a list of `Divisors`.

```
i2 : R = ZZ/32003[x_0..x_3];
i3 : Q = quadricSurface(R); X = cubicSurface(R);
i5 : Y = quarticSurfaceRational(R);
i6 : L = smoothDivisors(4,Q) | smoothDivisors(4,X) | smoothDivisors(4,Y);
i7 : netList L
     +------------------------------+
o7 = |{1, 3}                        |
     +------------------------------+
     |{2, 2}                        |
     +------------------------------+
     |{2, 1, 1, 0, 0, 0, 0}         |
     +------------------------------+
     |{3, 1, 1, 1, 1, 1, 0}         |
     +------------------------------+
     |{2, 0, 1, 1, 1, 1, 0, 0, 0, 0}|
     +------------------------------+
     |{3, 1, 1, 1, 1, 1, 1, 1, 0, 0}|
     +------------------------------+
```

Now we take the list of `Divisors` and turn them into `Curves`, and verify that the curves we get are indeed smooth and have the correct degrees and genera.

```
i8 : LC = apply(L,D->curve D);
i9 : netList apply(LC,C-> (degree C,genus C, isSmooth C))
     +------------+
o9 = |(4, 0, true)|
     +------------+
     |(4, 1, true)|
     +------------+
     |(4, 0, true)|
     +------------+
     |(4, 1, true)|
     +------------+
     |(4, 0, true)|
     +------------+
     |(4, 1, true)|
     +------------+
```

The method function `curve(ZZ,ZZ)` takes a pair of integers (d, g), generates divisors of degree d on all the surfaces, and selects one (if any) of genus g and returns it as a `Curve`.

```
i11 : C = curve(8,5);
i12 : degree C, genus C
o12 = (8, 5)
o12 : Sequence
```

The following code generated 608 curves up to degree 15 in around three minutes on a desktop with quad cores Intel i5-7400 CPU at 3.00GHz.

```
i2 : R = ZZ/101[x_0..x_3];
i3 : X = quadricSurface(R);
i4 : Y = cubicSurface(R);
i5 : Z = quarticSurfaceRational(R);
i6 : dmax = 15;
i7 : time LD = flatten apply(splice{1..dmax},d->
         smoothDivisors(d,X) | smoothDivisors(d,Y) | smoothDivisors(d,Z));
     -- used 2.3459 seconds
i8 : time LC = apply(LD,D -> curve D);
     -- used 185.549 seconds
i9 : #LC
o9 = 608
```

We use the method function `dgTable` to tally the number of curves by the degree (horizontal axis) and genus (vertical axis).

```
i10 : dgTable LC
o10 = 42 |  .  .  .  .  .  .  .  .  .  .  .  .  .  1
      41 |  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      40 |  .  .  .  .  .  .  .  .  .  .  .  .  .  1
      39 |  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      38 |  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      37 |  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      36 |  .  .  .  .  .  .  .  .  .  .  .  .  1  1
      35 |  .  .  .  .  .  .  .  .  .  .  .  .  1  .
      34 |  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      33 |  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      32 |  .  .  .  .  .  .  .  .  .  .  .  .  1  .
      31 |  .  .  .  .  .  .  .  .  .  .  .  .  .  1
      30 |  .  .  .  .  .  .  .  .  .  .  .  1  .  2
      29 |  .  .  .  .  .  .  .  .  .  .  .  .  .  1
      28 |  .  .  .  .  .  .  .  .  .  .  .  1  .  1
      27 |  .  .  .  .  .  .  .  .  .  .  .  .  1  3
      26 |  .  .  .  .  .  .  .  .  .  .  .  .  1  1
      25 |  .  .  .  .  .  .  .  .  .  .  1  .  1  3
      24 |  .  .  .  .  .  .  .  .  .  .  1  1  2  3
      23 |  .  .  .  .  .  .  .  .  .  .  .  .  1  3
      22 |  .  .  .  .  .  .  .  .  .  .  .  1  2  3
      21 |  .  .  .  .  .  .  .  .  .  .  1  1  2  5
      20 |  .  .  .  .  .  .  .  .  .  1  .  2  4  3
      19 |  .  .  .  .  .  .  .  .  .  .  1  1  2  3
      18 |  .  .  .  .  .  .  .  .  .  1  1  3  3  4
      17 |  .  .  .  .  .  .  .  .  .  .  1  2  3  4
      16 |  .  .  .  .  .  .  .  .  1  .  2  3  5  4
      15 |  .  .  .  .  .  .  .  .  1  1  3  2  3  6
      14 |  .  .  .  .  .  .  .  .  .  2  1  3  4  3
      13 |  .  .  .  .  .  .  .  .  .  2  3  3  3  2
      12 |  .  .  .  .  .  .  .  1  2  1  3  4  4  5
      11 |  .  .  .  .  .  .  .  .  1  2  3  2  3  2
      10 |  .  .  .  .  .  .  .  2  2  2  2  4  3  2
```

```
   9 |  .  .  .  .  .  .  1  1  1  3  5  4  4  4
   8 |  .  .  .  .  .  .  1  1  2  4  3  3  5  3
   7 |  .  .  .  .  .  .  1  1  4  3  4  3  3  4
   6 |  .  .  .  .  .  1  2  5  5  3  6  5  6  6
   5 |  .  .  .  .  .  1  4  2  3  4  5  4  6  5
   4 |  .  .  .  .  3  3  4  3  6  4  5  5  7  4
   3 |  .  .  .  .  3  3  3  4  5  4  8  5  5  5
   2 |  .  .  .  3  3  2  5  3  5  4  6  3  7  6
   1 |  .  1  3  2  3  3  5  3  5  4  5  5  8  5
   0 |  3  3  3  3  6  4  5  6  8  6  9  8  9 10
  ---+------------------------------------------
  g/d|  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

We close this section with a study of $\mathscr{H}^{\mathrm{sm}}_{8,5}$, the restricted Hilbert scheme of smooth curves of degree 8 and genus 5 in $\mathbb{P}^3$. It is known that $\mathscr{H}^{\mathrm{sm}}_{8,5}$ is irreducible; see for example [Ein 1986, Theorem 4]. In [Gruson and Peskine 1978, §4], we find a complete stratification of $\mathscr{H}^{\mathrm{sm}}_{8,5}$ into locally closed families A, B, C, D, E, as well as the Betti tables of each family. The following shows that the free resolutions for the families E, C, B, A (in this order) are indeed correct.

```
i13 : L = select(LC, C -> degree C == 8 and genus C == 5);
i14 : L / ideal / res / betti
                 0 1 2 3         0 1 2 3         0 1 2 3         0 1 2 3
o14 = {total: 1 6 8 3, total: 1 5 5 1, total: 1 4 4 1, total: 1 7 8 2}
             0: 1 . . .      0: 1 . . .      0: 1 . . .      0: 1 . . .
             1: . 1 . .      1: . . . .      1: . . . .      1: . . . .
             2: . . . .      2: . 1 . .      2: . 1 . .      2: . . . .
             3: . . . .      3: . 3 3 .      3: . 3 2 .      3: . 7 8 2
             4: . . . .      4: . 1 2 1      4: . . 2 1
             5: . 5 8 3
o14 : List
```

We note that curves in family D are missing from our program because they do not lie on any of the surfaces we constructed. In fact, they all lie on the ruled cubic surface with a double line. In the upcoming 2.0 version of the package, we will include curves on this surface and much more.

3. ACM curves in projective three-space.

In this section we consider arithmetically Cohen–Macaulay (ACM) curves in $\mathbb{P}^3$. The literature on ACM curves in $\mathbb{P}^3$ is very rich and we do not aim to give a survey in this article. We shall illustrate the stratification of Hilbert schemes of ACM curves by Betti tables.

3.1. *Hilbert functions of ACM curves.*

Let $\mathscr{H}^{\mathrm{CM}}$ denote the points of the Hilbert scheme (without fixing a Hilbert polynomial) of curves in $\mathbb{P}^3$ that correspond to ACM curves. Ellingsrud [1975] shows that $\mathscr{H}^{\mathrm{CM}}$ is an open smooth subscheme of the Hilbert scheme. Let $\mathscr{H}^{\mathrm{CM}}_{H}$ denote the points that correspond to ACM curves with Hilbert function H, then we have a stratification

$$\mathscr{H}^{\mathrm{CM}} = \bigsqcup_H \mathscr{H}_H^{\mathrm{CM}}.$$

Furthermore, it is shown in [Ellingsrud 1975] that the spaces $\mathscr{H}_H^{\mathrm{CM}}$ form disjoint irreducible connected components of $\mathscr{H}^{\mathrm{CM}}$.

What are all the possible Hilbert functions of ACM curves in $\mathbb{P}^3$?

There are many notions that encode information which is equivalent to that encoded by the Hilbert function, e.g., *numerical characters* as in [Gruson and Peskine 1978, §2], *postulation characters* as in [Martin-Deschamps and Perrin 1990, I.2] and *h-vectors* as in [Migliore 1998, §1.4]. For computational reasons that we will explain later, we will use the postulation character of a curve C, which is defined to be the negative of the third discrete difference of its Hilbert function.

A theorem of Gruson and Peskine [1978, §2] says that the postulation characters of ACM curves are exactly the *positive characters*. A positive character is a function $\gamma : \mathbb{Z} \to \mathbb{Z}$ of finite support, such that the following hold:

(1) $\sum_n \gamma(n) = 0$,

(2) $\gamma(n) = 0$ for $n < 0$,

(3) $\gamma(0) = -1$,

(4) If we set $s = s(\gamma) = \inf\{n|\gamma(n) \neq -1\}$, then $\gamma(s) \geq 0$,

(5) $\gamma(n) \geq 0$ for $n \geq s$.

The degree of a positive character γ is defined to be $\sum_n n\gamma(n)$. If γ is the postulation character of a curve C, then the degree of γ is the degree of C, and $s(\gamma)$ is the least degree surface C lies on.

Given d and s, the enumeration of all positive characters becomes a familiar problem: what are the different ways to make $d - \sum_{n<s} n$ cents in total using s coins where each coin can have value n for every $n \geq s$? Macaulay2 solves this efficiently by creating a bigraded ring and enumerating monomials of a fixed bidegree. The method function `positiveChars(ZZ,ZZ)` does exactly that, and enumerates all positive characters with a given degree d and s. By looping s from 1 to $d-1$, the method function `positiveChars(ZZ)` then enumerates all positive characters of a given degree d.

```
i2 : positiveChars(6,2)
o2 = {{-1, -1, 1, 0, 0, 1}, {-1, -1, 0, 1, 1}}
o2 : List
i3 : positiveChars(6)
o3 = {{-1, -1, 1, 0, 0, 1}, {-1, -1, 0, 1, 1}, {-1, -1, -1, 3}}
o3 : List
```

We can produce all 121 positive characters up to degree 16 within 0.05 seconds. The efficiency is one of the main reasons why we chose the postulation character to represent the Hilbert function.

```
i4 : time L = flatten apply(15, d -> positiveChars(d+1));
     -- used 0.0492106 seconds
i5 : #L
o5 = 121
```

3.2. ***Betti tables of ACM curves.*** We define a further stratification

$$\mathscr{H}_H^{\mathrm{CM}} = \bigsqcup_B \mathscr{H}_{H,B}^{\mathrm{CM}},$$

where $\mathscr{H}_{H,B}^{\mathrm{CM}}$ consists of points that correspond to ACM curves with Hilbert function H and Betti table B.

What are all the Betti tables B of an ACM curve in $\mathbb{P}^3$?

A necessary and sufficient condition for a Betti table B of a homogeneous ideal to be that of an ACM curve in $\mathbb{P}^3$ is the following: B has $n+1$ generators of degrees $a_1 \geq \cdots \geq a_{n+1}$ and n syzygies of degrees $b_1 \geq \cdots \geq b_n$ and no higher syzygies, such that the diagonal entries of the matrix $M = (a_i - b_j)_{i,j}$ are positive. See, for example, [Eisenbud 2005, Propositions 3.8 and 3.14]. We call M the Hilbert–Burch degree matrix.

For a given Hilbert function H, let $\mathfrak{B}_H$ denote the set of Betti tables B such that $\mathscr{H}_{H,B}^{\mathrm{CM}}$ is nonempty. We partially order $\mathfrak{B}_H$ by the number of generators (equivalently syzygies) and write $B >_n B'$ if B has n more generators than B'. Since the Hilbert function H is fixed, the fourth discrete difference $\Delta^4 H$ is equal to the alternating sum of the Betti numbers $\sum_i (-1)^i B_{i,n}$ for every $B \in \mathfrak{B}_H$. Therefore $B >_1 B'$ if and only if B is obtained from B' by adding one generator and one syzygy of the same degree, and $B >_n B'$ if and only if B can be obtained from B' by n successive additions of one generator and one syzygy of the same degree. By the previous paragraph, if we remove a generator and a syzygy of the same degree from some $B \in \mathfrak{B}_H$ to obtain B', then $B' \in \mathfrak{B}_H$ also. Thus B is minimal in $\mathfrak{B}_H$ if and only if no generator and syzygy of B share the same degree, but in this case B is determined uniquely by $\Delta^4 H = \sum_i (-1)^i B_{i,n}$. In conclusion: $\mathfrak{B}_H$ has a smallest element denoted by $\mathfrak{b}$, and every $B \in \mathfrak{B}_H$ can be obtained from $\mathfrak{b}$ by successively adding one generator and one syzygy of the same degree (although not all ways of doing so yield a Betti table in $\mathfrak{B}_H$).

The main theorem of [Ellingsrud 1975] states that the downward closed strata $\bigsqcup_{B \leq B'} \mathscr{H}_{H,B}^{\mathrm{CM}}$ is open and irreducible (hence smooth) in $\mathscr{H}_H^{\mathrm{CM}}$. Further, for two Betti tables $B, B' \in \mathfrak{B}_H$, we have $B \geq B'$ if and only if $\overline{\mathscr{H}_{H,B'}^{\mathrm{CM}}} \supset \mathscr{H}_{H,B}^{\mathrm{CM}}$. In other words, the partial order on $\mathfrak{B}_H$ corresponds to the partial order of specialization among the strata $\mathscr{H}_{H,B}^{\mathrm{CM}}$.

The method function `generalACMBetti(List)` takes a postulation character γ corresponding to the Hilbert function H and returns the smallest element $\mathfrak{b}$ of $\mathfrak{B}_H$.

```
i2 : generalACMBetti {-1,-1,-1,2,1}
            0 1 2
o2 = total: 1 3 2
         0: 1 . .
         1: . . .
         2: . 3 1
         3: . . 1
o2 : BettiTally
```

The method function `specializeACMBetti(BettiTally)` takes a Betti table $B \in B_H$ and returns the list of all $B' \in \mathfrak{B}_H$ such that $B' >_1 B$.

```
i3 : specializeACMBetti oo
             0 1 2
o3 = {total: 1 4 3}
          0: 1 . .
          1: . . .
          2: . 3 2
          3: . 1 1
o3 : List
```

Applying this iteratively on $\mathfrak{b}$, we exhaust all of $\mathfrak{B}_H$ in layers. The method function `allACMBetti(List)` takes a postulation character, and returns all Betti tables of ACM curves having that character. The following are all the Betti tables of ACM curves of degree 8. Each row corresponds to a distinct Hilbert function, and along each row the Betti tables are more special as we go from left to right. Note that these outputs are only Betti tables, no curves are produced yet.

```
i5 : netList (positiveChars(6) / allACMBetti)
     +------------+------------+
     |       0 1 2|            |
o5 = |total: 1 3 2|            |
     |    0: 1 . .|            |
     |    1: . 2 1|            |
     |    2: . . .|            |
     |    3: . . .|            |
     |    4: . 1 1|            |
     +------------+------------+
     |       0 1 2|       0 1 2|
     |total: 1 2 1|total: 1 3 2|
     |    0: 1 . .|    0: 1 . .|
     |    1: . 1 .|    1: . 1 .|
     |    2: . 1 .|    2: . 1 1|
     |    3: . . 1|    3: . 1 1|
     +------------+------------+
     |       0 1 2|            |
     |total: 1 4 3|            |
     |    0: 1 . .|            |
     |    1: . . .|            |
     |    2: . 4 3|            |
     +------------+------------+
```

The method function `degreeMatrix(BettiTally)` takes the Betti table of an ACM curve, and returns its Hilbert–Burch degree matrix. The method function `randomDeterminantalIdeal(Ring,Matrix)` computes the ideal of maximal minors of a matrix of random forms with specified degrees. Together, these method functions allow us to generate ACM curves. The following code generates an ACM curve for each of the Betti table given above, and shows that they do indeed have the predicted Betti tables.

```
i7 : L =  positiveChars(6) / allACMBetti;
i8 : netList apply(L, H -> H / (B ->
     betti res randomDeterminantalIdeal(ZZ/101[x,y,z,w],degreeMatrix B)))
     +------------+------------+
     |       0 1 2|            |
o8 = |total: 1 3 2|            |
     |    0: 1 . .|            |
     |    1: . 2 1|            |
     |    2: . . .|            |
     |    3: . . .|            |
     |    4: . 1 1|            |
     +------------+------------+
     |       0 1 2|       0 1 2|
     |total: 1 2 1|total: 1 3 2|
     |    0: 1 . .|    0: 1 . .|
     |    1: . 1 .|    1: . 1 .|
     |    2: . 1 .|    2: . 1 1|
     |    3: . . 1|    3: . 1 1|
     +------------+------------+
     |       0 1 2|            |
     |total: 1 4 3|            |
     |    0: 1 . .|            |
     |    1: . . .|            |
     |    2: . 4 3|            |
     +------------+------------+
```

Here is an example of two layers of specializations. The second and the third families are incomparable, and they both specialize to the fourth family.

```
i9 : gamma = (positiveChars(9))#5
o9 = {-1, -1, -1, 1, 1, 1}
o9 : List
i10 : allACMBetti gamma
              0 1 2          0 1 2          0 1 2          0 1 2
o10 = {total: 1 2 1, total: 1 3 2, total: 1 3 2, total: 1 4 3}
          0: 1 . .       0: 1 . .       0: 1 . .       0: 1 . .
          1: . . .       1: . . .       1: . . .       1: . . .
          2: . 2 .       2: . 2 1       2: . 2 .       2: . 2 1
          3: . . .       3: . 1 .       3: . . 1       3: . 1 1
          4: . . 1       4: . . 1       4: . 1 1       4: . 1 1
o10 : List
```

We end this section with a short comment on smoothness. When does $\mathscr{H}_{H,B}^{\mathrm{CM}}$ have a point that corresponds to a smooth ACM curve? The answer is given by [Geramita and Migliore 1989, Proposition 1.3]: if and only if the Hilbert–Burch degree matrix has positive upper diagonal. We leave it as an exercise to the reader to check which of the four families above have a smooth ACM curve.

4. Minimal curves in an even liaison class.

The Hartshorne–Rao module (or deficiency module) of a curve C is defined to be the graded module

$$M_C := H^1_*(\mathscr{I}_C) = \bigoplus_{n\in\mathbb{Z}} H^0(\mathscr{I}_C(n))$$

over the polynomial ring $S = H^0_*(\mathscr{O}_{\mathbb{P}^3})$. Since the curves we consider are locally Cohen–Macaulay and equidimensional, one can show that M_C is of finite length.

The method `raoModule(Ideal)` computes the Hartshorne–Rao module from the ideal of a curve using local duality. We compute a free resolution $F_\bullet$ of S/I_C, then dualize and take the cokernel of the last term to compute $\mathrm{Ext}^3_S(S/I_C, S)$. Then we resolve $\mathrm{Ext}^3_S(S/I_C, S)$ by $G_\bullet$ and again dualize and take the cokernel of the last term to obtain M_C. The following is an example of the rational quartic curve.

```
ii78 : I = monomialCurveIdeal(R,{1,3,4});
oo78 : Ideal of R
ii79 : betti res I
              0 1 2 3
oo79 = total: 1 4 4 1
           0: 1 . . .
           1: . 1 . .
           2: . 3 4 1
oo79 : BettiTally
ii80 : M = raoModule I
oo80 = cokernel {1} | w -z y -x |
                                1
oo80 : R-module, quotient of R
ii81 : betti res M
              0 1 2 3 4
oo81 = total: 1 4 6 4 1
           1: 1 4 6 4 1
oo81 : BettiTally
```

Rao [1978/79] made the beautiful discovery that there is a bijection between the even liaison classes of curves and isomorphism classes of finite length modules up to a shift in grading. We say a curve C is minimal if for every curve D in the even liaison class of C, we have $M_D \cong M_C[h]$ for some $h \geq 0$. It turns out that a minimal curve also has the minimal degree and genus among all curves in its even liaison class, and they are unique up to deformation with constant cohomology. Furthermore, all curves in the even liaison class can be obtained from C by finitely many basic double links and a deformation with constant cohomology. In particular, the possible degrees and genera of all curves in the even liaison class can be computed from the minimal curve C alone. This vastly generalizes the case of ACM curves where $H^1_*(\mathscr{I}_C) = 0$. We refer the readers to [Martin-Deschamps and Perrin 1990] for details.

Given a finite length graded S-module M, we wish to construct a minimal curve C such that $M_C \cong M[h]$ for some smallest possible integer h. A suitable algorithm is outlined in [Martin-Deschamps and Perrin 1990] and improved by [Guarrera et al. 1997]. We give a simplified account here. Consider the short exact sequence of S-modules

$$0 \to S/I_C \to H^0_*(\mathscr{I}_C) \to M_C \to 0.$$

If $F_\bullet$ and $G_\bullet$ are minimal free resolutions of S/I_C and $H^0_*(\mathscr{I}_C)$, respectively, and $\alpha : F_\bullet \to G_\bullet$ is an induced map, then the mapping cone $C(\alpha)_\bullet$ is a resolution of M_C. By [Martin-Deschamps and Perrin 1990, Theorems 3.7 and 4.1], if C is a minimal curve, then $C(\alpha)_\bullet$ is actually minimal. Since $H^0_*(\mathscr{I}_C)$ has depth 2, it has projective dimension 2 by the Auslander–Buchsbaum formula. We see that $F_2 \cong L_4$, $F_1 \cong L_3$ and F_0 is a summand of L_2.

We turn this observation around, and start with a minimal free resolution $L_\bullet$ of a finite length module M and ask: what are the conditions on the projections π such that $\operatorname{coker} d = \operatorname{coker}(\pi \circ d_3)$ is the ideal of a curve in $\mathbb{P}^3$ after shifting by an integer h?

The Buchsbaum–Eisenbud criterion of exactness tells us that if the lower complex in

$$\begin{tikzcd}
0 \arrow[r] & L_4 \arrow[r] \arrow[d, equal] & L_3 \arrow[r, "d_3"] \arrow[d, equal] & L_2 \arrow[r] \arrow[d, "\pi"] & \Omega^2 M \arrow[r] \arrow[d] & 0 \\
0 \arrow[r] & L_4 \arrow[r] & L_3 \arrow[r, "d"] & P \arrow[r] & \operatorname{coker} d \arrow[r] & 0
\end{tikzcd}$$

is a free resolution of an ideal (up to shift), then $\operatorname{rk} P = \operatorname{rk} L_3 + 1 - \operatorname{rk} L_4 = \operatorname{rk} d + 1$ and grade $I(d) \geq 2$. It turns out that this is also sufficient. We rephrase the grade $I(d) \geq 2$ condition in terms of the ideal of maximal nonvanishing minors: this is if and only if $I(d)$ is not contained in any principal ideal, i.e., $\operatorname{rk} d \otimes_S S/(f) = \operatorname{rk} d$ for all $f \in S$. Since the alternating sum of the degrees of the free modules must be zero for the free resolution of a curve by the Herzog–Kühl equations, we can easily determine h whenever the degrees of P are known. To summarize, we are looking for projections π satisfying the above conditions such that the resulting shift h is as small as possible.

The key observation of [Guarrera et al. 1997] is that the rank and the rank in codimension 1 of a matrix M over a PID can be easily read off from its Smith normal form. Furthermore, if we evaluate the matrix d in a PID $k[T]$, then for a general choice of evaluations $k[x_0, \ldots, x_3] \to k[t]$, the rank and rank in codimension 1 of d are preserved. The method function `minimalCurve(Module)` implements this method to find a projection π corresponding to a minimal curve.

Let us generate a minimal curve in the even liaison class of the rational quartic.

```
ii82 : J = minimalCurve M;
oo82 : Ideal of R
ii83 : betti res J
              0 1 2 3
oo83 = total: 1 4 4 1
           0: 1 . . .
           1: . 4 4 1
oo83 : BettiTally
```

Indeed, the rational quartic curve has Hartshorne–Rao module k concentrated in degree 1, and lies on a smooth quadric with three other cubic generators. The complete intersection of the quadric with a general linear combination of the cubic generators is a divisor of type $(3, 3)$ on the quadric surface. Since the rational quartic is of type $(1, 3)$, the residual would have type $(2, 0)$ — the union of two skew lines. The latter has Hartshorne–Rao module k concentrated in degree 0, and is obviously of minimal degree in its even liaison class.

Finally, here is a remark about the function `minimalCurve(Module)`. Once the matrix d is computed, there are two ways to obtain the equations of the ideal I_C: the first is to compute the kernel of the transposition of d, the second is to take maximal minors and saturate by the irrelevant ideal. Unfortunately, both would take a long time if d is a large matrix. To get partial information, the method function `minimalCurveBetti(Module)` outputs the Betti table of a minimal curve corresponding to a finite length module without explicitly computing the ideal of the curve.

ACKNOWLEDGEMENTS. The author thanks F. Schreyer and M. Stillman for initiating the project and R. Hartshorne for guidance. The author also thanks D. Eisenbud for supervision and support.

SUPPLEMENT. The online supplement contains version 1.0 of SpaceCurves.

REFERENCES.

[Ein 1986] L. Ein, "Hilbert scheme of smooth space curves", *Ann. Sci. École Norm. Sup.* (4) **19**:4 (1986), 469–478. MR Zbl

[Eisenbud 2005] D. Eisenbud, *The geometry of syzygies: a second course in commutative algebra and algebraic geometry*, Graduate Texts in Mathematics **229**, Springer, 2005. MR Zbl

[Ellingsrud 1975] G. Ellingsrud, "Sur le schéma de Hilbert des variétés de codimension 2 dans $\mathbf{P}^e$ à cône de Cohen–Macaulay", *Ann. Sci. École Norm. Sup.* (4) **8**:4 (1975), 423–431. MR Zbl

[Geramita and Migliore 1989] A. V. Geramita and J. C. Migliore, "Hyperplane sections of a smooth curve in $\mathbf{P}^3$", *Comm. Algebra* **17**:12 (1989), 3129–3164. MR Zbl

[Gruson and Peskine 1978] L. Gruson and C. Peskine, "Genre des courbes de l'espace projectif", pp. 31–59 in *Algebraic geometry* (Tromsø, 1977), edited by L. D. Olsen, Lecture Notes in Math. **687**, Springer, Berlin, 1978. MR Zbl

[Gruson and Peskine 1982] L. Gruson and C. Peskine, "Genre des courbes de l'espace projectif, II", *Ann. Sci. École Norm. Sup.* (4) **15**:3 (1982), 401–418. MR Zbl

[Guarrera et al. 1997] S. Guarrera, A. Logar, and E. Mezzetti, "An algorithm for computing minimal curves", *Arch. Math.* (*Basel*) **68**:4 (1997), 285–296. MR Zbl

[Halphen 1882] G. Halphen, "Mémoire sur la classification des courbes gauches algébriques", *J. Éc. Poly.* **52** (1882), 1–200.

[Hartshorne 1977] R. Hartshorne, *Algebraic geometry*, Graduate Texts in Mathematics **52**, Springer, 1977. MR Zbl

[Hartshorne 1982] R. Hartshorne, "Genre de courbes algébriques dans l'espace projectif (d'après L. Gruson et C. Peskine)", exposé 592] 301–313 in *Séminaire Bourbaki,* 1981/1982, Astérisque, Soc. Math. France, Paris, 1982. MR Zbl

[Martin-Deschamps and Perrin 1990] M. Martin-Deschamps and D. Perrin, *Sur la classification des courbes gauches*, Astérisque **184-185**, 1990. MR Zbl

[Migliore 1998] J. C. Migliore, *Introduction to liaison theory and deficiency modules*, Progress in Mathematics **165**, Birkhäuser, Boston, 1998. MR Zbl

[Rao 1978/79] P. Rao, "Liaison among curves in $\mathbf{P}^3$", *Invent. Math.* **50**:3 (1978/79), 205–217. MR Zbl

RECEIVED: 19 Oct 2017 REVISED: 10 Apr 2018 ACCEPTED: 18 May 2018

MENGYUAN ZHANG:
myzhang@berkeley.edu
Department of Mathematics, University of California, Berkeley, United States

JOURNAL OF SOFTWARE FOR ALGEBRA AND GEOMETRY vol 8, no 1, 2018