THE QUILLENSUSLIN PACKAGE FOR MACAULAY2

BRETT BARWICK AND BRANDEN STONE

Abstract. The *QuillenSuslin* package for *Macaulay2* provides the ability to compute a free basis for a projective module over a polynomial ring with coefficients in $\mathbb{Q}, \mathbb{Z}$, or $\mathbb{Z}/p$ for a prime integer p. A brief description of the underlying algorithm and the related tools are given.

1. Introduction

In 1955, J-P. Serre posed the following question: do there exist finitely generated projective $k[x_1, \ldots, x_n]$ modules, with k a field, which are not free? [Ser55] This question was known as "Serre's Problem" and the question in its full generality remained open for 21 years until it was resolved independently by D. Quillen and A. A. Suslin in 1976, resulting in the following well-known theorem.

Theorem 1 (Quillen-Suslin, 1976 [Qui76, Sus76])**.** *Let $S = k[x_1, \ldots, x_n]$, with k a field. Then every finitely generated projective S-module is free.*

However, the proofs given were not entirely constructive, and it was not until the early 1990's that papers such as [FG90, LS92, LW00] began giving fully constructive versions of the proof. In 1992, A. Logar and B. Sturmfels [LS92] published the algorithmic proof of the Quillen-Suslin Theorem that forms the basis for the methods in *QuillenSuslin*. In their paper, Logar and Sturmfels describe, via the technique of completion of unimodular rows, how to construct a free generating set for a projective module over $\mathbb{C}[x_1, \ldots, x_n]$. One can extend these constructive techniques to work over more general coefficient rings such as $\mathbb{Q}, \mathbb{Z}$, and $\mathbb{Z}/p\mathbb{Z}$, for p a prime integer. Descriptions of some of these more general techniques, along with applications to areas such as systems control theory, can be found in [Fab09, Ch. 2] and [FQ07, Ch. 3], and the algorithms given in these papers were used to create a similar QuillenSuslin package [Fab] for the computer algebra system *Maple*. We have implemented these algorithms, with some modifications, in our *QuillenSuslin* package for *Macaulay2* [GS]. In the next section we will give some preliminary definitions and results that reduce the statement of the Quillen-Suslin Theorem to a more concrete matrix theoretic problem concerning the completion of unimodular rows over polynomial rings to square invertible matrices.

2. Preliminaries

In this section, R will denote a commutative ring and M will denote a finitely generated R-module. We say that M is a projective R-module if it is a direct summand of a free module. Similarly, we define a slightly stronger notion by saying

Date: May 10, 2013.
2010 *Mathematics Subject Classification.* Primary: 13P99; Secondary: 13C10.
The first author was partially supported by NSA Grant No. H98230-10-1-0361.

that M is stably free if there exists some $m \geq 0$ such that $M \oplus R^m$ is free. A module M is stably free if and only if it is isomorphic to the kernel of a surjective R-linear map $\phi : R^n \to R^m$ for some $m \leq n$. Since ϕ surjects onto a free module, we know that this map splits and admits a right inverse $\psi : R^m \to R^n$ so that $\phi\psi = \mathrm{id}_{R^m}$. Therefore a matrix representing ϕ is right invertible, and we call such a right invertible matrix over R unimodular. Using this terminology, it is not difficult to show that the Quillen-Suslin Theorem as stated above is equivalent to the following matrix theoretic statement about unimodular matrices.

Theorem 2 (Quillen-Suslin, restatement [LS92, Theorem 1.1])**.** *Let $S = R[x_1, \ldots, x_n]$ with R a principal ideal domain and let $U \in Mat_{m\times n}(R)$ be a unimodular matrix over S with $m \leq n$. Then there exists a unimodular matrix $V \in Mat_{n\times n}(S)$ such that*

$$UV = \left[\begin{array}{cccc|ccc} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \end{array}\right].$$

A matrix V that satisfies the properties in the above theorem is said to solve the unimodular matrix problem for U. Given such a V, the first m rows of the invertible matrix V^{-1} are the same as the original matrix U. Therefore we see that proving the Quillen-Suslin Theorem is equivalent to showing that any unimodular matrix can be completed to a square invertible matrix over the polynomial ring. As it turns out, it suffices to show that the unimodular row problem can be solved, that is, Theorem 2 holds for unimodular row vectors [LS92]. For more details concerning this equivalent formulation of the Quillen-Suslin Theorem, we refer the interested reader to the excellent book of Lam [Lam06].

3. The Logar-Sturmfels Algorithm

Before describing the general algorithm, we mention that *QuillenSuslin* contains several shortcut methods as described in [Fab09, Sect. 2.2]. These methods allow us to quickly solve the unimodular row problem for a row satisfying certain properties; often allowing us to avoid the worst-case general algorithm. These shortcut methods are automatically used as soon as they are applicable during the methods `computeFreeBasis`, `completeMatrix`, `qsAlgorithm`, and `qsIsomorphism`.

The main idea behind the Logar-Sturmfels algorithm is to iteratively reduce the number of variables involved in a unimodular row $\boldsymbol{f}$ one by one, eventually obtaining a unimodular row $\tilde{\boldsymbol{f}}$ over the coefficient ring, which is a PID. We can then use a simple algorithm based on the Smith normal form of $\tilde{\boldsymbol{f}}$ to construct a final unimodular matrix U so that $\tilde{\boldsymbol{f}}U = [1\ 0\ \cdots\ 0]$. Multiplying together all of the matrices used during the process, one can construct a unimodular matrix that solves the unimodular row problem for the original row $\boldsymbol{f}$.

The process of eliminating a variable from a unimodular matrix is organized into three main steps: the normalization step, the "local loop", and the patching step. Below is a brief description of each step, as well as a demonstration of the corresponding commands in the *QuillenSuslin* package. We will work over the polynomial ring $S = \mathbb{Z}[x, y]$ and consider the unimodular row $\boldsymbol{f} = [x^2\,,\ 2y+1\,,\ x^5y^2+y]$ over S.

```
i1 : loadPackage "QuillenSuslin";
i2 : S = ZZ[x,y];

i3 : f = matrix {{x^2,2*y+1,x^5*y^2+y}}
o3 = | x2 2y+1 x5y2+y |
             1       3
o3 : Matrix S  <--- S
```

We can use the command `isUnimodular` to check that this row is indeed unimodular over S.

```
i4 : isUnimodular f
o4 = true
```

In order to eliminate the variable y from this unimodular row, we will construct a unimodular matrix U so that $\boldsymbol{f}U$ is the same as $\boldsymbol{f}$ with y replaced by 0. We first demonstrate the normalization step.

3.1. **Normalization Step.** Since Horrocks' Theorem (see Theorem 4 below) requires a monic polynomial, we must first construct a unimodular matrix U_1 and an invertible change of variables $x_i \leftrightarrow X_i$ so that the first entry of $\boldsymbol{f}U_1$ is monic in X_n. The normalization step is based on the following result, which has been slightly restated for our purposes.

Lemma 3 ([VS76, Lemma 10.6])**.** *Let R be a Noetherian ring, n and m natural numbers, $S = R[x_1, \ldots, x_n]$, $m \geq \dim R + 2$, and $\boldsymbol{f} = [f_1\ f_2\ \cdots\ f_m]$ a unimodular row over S. Then there exists an $m \times m$ unimodular matrix U over S, and an invertible change of variables $x_i \leftrightarrow X_i$, so that after applying the change of variables the first entry of $\boldsymbol{f}U$ is monic in X_n when viewed as a polynomial in $R[X_1, \ldots, X_{n-1}][X_n]$.*

A constructive version of this result is implemented in the method `changeVar`, and is used as in the following example.

```
i5 : (U1,subs,invSubs) = changeVar(f,{x,y})
o5 = (| 1 0 0 |, | y x |, | y x |)
      | 0 1 0 |
      | 0 0 1 |
o5 : Sequence

i6 : f = sub(f*U1,subs)
o6 = | y2 2x+1 x2y5+x |
             1       3
o6 : Matrix S  <--- S
```

Notice that since the first entry of the row $\boldsymbol{f}$ was already monic in x, the method simply returned a permutation of the variables interchanging x and y so that the first entry of the new row would be monic in y. Now that the first entry of the row is monic in the variable we are trying to eliminate, we may proceed to the "local loop."

3.2. **Local Loop.** The purpose of the local loop is to compute a collection of *local solutions* to the unimodular row problem for $\boldsymbol{f}$. The local loop is based on the following result of Horrocks.

Theorem 4 (Horrocks, [Rot09, Prop. 4.98])**.** *Consider the polynomial ring $B[y]$, where B is a local ring, and let $\boldsymbol{f} = [f_1\ f_2\ \cdots\ f_m]$ be a unimodular row over $B[y]$. If some f_i is monic in y, then there exists a unimodular $m \times m$ matrix U over $B[y]$ so that $\boldsymbol{f}U = [1\ 0\ \cdots\ 0]$.*

In order to eliminate the last variable x_n in a unimodular row over a polynomial ring $R[x_1, \dots, x_n]$, the local loop proceeds in the following way: First set $I = (0)$ in $A = R[x_1, \dots, x_{n-1}]$. Now while $I \neq A$, the i^{th} iteration of the loop is

(1) Find a maximal ideal $\mathfrak{m}_i$ in A containing I.
(2) Apply Horrocks' Theorem to the row $\boldsymbol{f}$, viewed as a unimodular row over $A_{\mathfrak{m}_i}[x_n]$, to find a unimodular matrix L_i over $A_{\mathfrak{m}_i}[x_n]$ that solves the unimodular row problem for $\boldsymbol{f}$ (we call this L_i a local solution to the unimodular row problem for $\boldsymbol{f}$).
(3) Let d_i denote the common denominator for all of the elements in the matrix U_i.
(4) Set $I = I + (d_i)$.

If $I \neq A$, then we repeat the loop. Otherwise we are able to stop and go on to the patching step. Notice that since we are creating a strictly ascending chain of ideals $(d_1) \subset (d_1, d_2) \subset \cdots$ in the Noetherian ring A, this loop must terminate in a finite number of steps with $(d_1, \dots, d_k) = A$ for some integer k.

In our example, we use the method `getMaxIdeal` to first find an arbitrary maximal ideal in $\mathbb{Z}[x]$, and we set $\mathfrak{m}_1 = (2, x)$. Using the method `horrocks`, we can compute a unimodular matrix L_1 over $\left(\mathbb{Z}[x]_{(2,x)}\right)[y]$ so that $\boldsymbol{f}L_1 = [1\ 0\ 0]$.

```
i7 : A = ZZ[x];

i8 : m1 = getMaxIdeal(ideal(0_A),{x})
o8 = ideal (2, x)
o8 : Ideal of A

i9 : L1 = horrocks(f,y,m1)
o9 = | 0         1          0                 |
     | 1/(2x+1)  -y2/(2x+1) (-x2y5-x)/(2x+1)  |
     | 0         0          1                 |
                       3                3
o9 : Matrix (frac S)  <--- (frac S)
```

Since $d_1 = 2x+1$ is a common denominator for the entries of L_1 and $(2x+1) \neq \mathbb{Z}[x]$, we use `getMaxIdeal` again to find a maximal ideal containing $2x + 1$, and we set $\mathfrak{m}_2 = (3, x - 1)$. We use `horrocks` a second time to compute a new local solution L2 with common denominator $d_2 = x$.

```
i10 : m2 = getMaxIdeal(sub(ideal(2*x+1),A),{x})
o10 = ideal (x - 1, 3)
o10 : Ideal of A

i11 : L2 = horrocks(f,y,m2)
o11 = | -xy3 xy5+1 2x2y3+xy3 |
      | 0    0     1         |
      | 1/x  -y2/x (-2x-1)/x |
```

```
                        3               3
o11 : Matrix (frac S)  <--- (frac S)


i12 : sub(ideal(2*x+1,x),S) == ideal(1_S)
o12 = true
```

Since $(d_1, d_2) = (2x+1, x) = \mathbb{Z}[x]$, we are able to exit the local loop and proceed to the patching step.

3.3. **Patching Step.** Loosely speaking, the patching step involves multiplying slight variations of the local solutions $L_1, \ldots, L_k$ together in a clever way so that the product U is a unimodular matrix over the polynomial ring $R[x_1, \ldots, x_n]$ and multiplying $\boldsymbol{f}$ times U is equivalent to evaluating $\boldsymbol{f}$ when $x_n = 0$, thereby eliminating one of the variables in the row $\boldsymbol{f}$. For more details, see [LS92, pg. 235].

Following along with our example, we use the method `patch` applied to our list $\{L_1, L_2\}$ of local solutions and we specify that y is the variable that we want to eliminate.

```
i13 : U = patch({L1,L2},y)
o13 = | -32x6y5+1                    0 8x5y3              |
      | 16x5y7-8x4y7+4x3y7+2xy2-y2 1 -4x4y5+2x3y5-x2y5 |
      | -4xy2                        0 1                  |
              3       3
o13 : Matrix S  <--- S

i14 : f*U
o14 = | 0 2x+1 x |
              1       3
o14 : Matrix S  <--- S
```

We can see that multiplying the row $\boldsymbol{f}$ times the unimodular matrix `U` is equivalent to evaluating $\boldsymbol{f}$ when $y = 0$ (keeping in mind that the variables x and y were interchanged during the normalization step).

4. Core methods in the QuillenSuslin package

The method `qsAlgorithm` automates all of the above computations for computing a solution to the unimodular matrix problem, and automatically applies the shortcut methods in [Fab09] when possible. We demonstrate the use of `qsAlgorithm` by finding a solution to the unimodular row problem for the row $\boldsymbol{f} = [y^2\,,\, 2x+1\,,\, x^2y^5+x]$ given earlier.

```
i15 : U = qsAlgorithm f
o15 = | 2x3y2 -2x5y2+1 -2x8y4-2x3y3 |
      | 1     -x2      -x5y2-y      |
      | -2    2x2      2x5y2+2y+1   |
              3       3
o15 : Matrix S  <--- S

i16 : det U
o16 = -1
```

```
o16 = S
i17 : f*U
o17 = | 1 0 0 |
              1       3
o17 : Matrix S  <--- S
```

The package also contains a method `completeMatrix` that completes a unimodular matrix over a polynomial ring to a square invertible matrix. Again we demonstrate its use on the unimodular row $\boldsymbol{f}$.

```
i18 : C = completeMatrix f
o18 = {0} | y2 2x+1   x2y5+x |
      {2} | 1  -2x2y3 0      |
      {7} | 0  2      1      |
              3       3
o18 : Matrix S  <--- S

i19 : det C
o19 = -1
o19 : S
```

The previous two methods, `qsAlgorithm` and `completeMatrix`, also work over Laurent polynomial rings of the form $k[x_1^{\pm 1}, \ldots, x_n^{\pm 1}]$ with $k = \mathbb{Q}$ or $\mathbb{Z}/p\mathbb{Z}$ for p a prime integer. The algorithm in the Laurent polynomial case is due to Park and is described in [Par04, p. 215].

Finally, we give an example to demonstrate the method `computeFreeBasis` that computes a free generating set for a projective module. We define $K = \ker \boldsymbol{f}$, which we can check is a projective $\mathbb{Z}[x, y]$-module by using the command `isProjective`.

```
i20 : K = ker f
o20 = image {2} | 2y+1 2x3y3+x3y2 2x5y2-1 -x5y2-y |
            {1} | -x2  y          x2      0       |
            {7} | 0    -2y-1      -2x2    x2      |
                                3
o20 : R-module, submodule of R

i21 : isProjective K
o21 = true

i22 : mingens K
o22 = {2} | -2y-1 2x3y3+x3y2 x5y2+y |
      {1} | x2    y          0      |
      {7} | 0     -2y-1      -x2    |
              3       3
o22 : Matrix R  <--- R

i23 : syz mingens K
o23 = {3} | y     |
      {8} | -x2   |
      {9} | 2y+1  |
              3       1
o23 : Matrix R  <--- R
```

```
i24 : phi = qsIsomorphism K
o24 = {3} | 0 0 |
      {8} | 1 0 |
      {9} | 0 1 |
      {9} | 0 0 |
o24 : Matrix

i25 : source phi
        2
o25 = R
o25 : R-module, free

i26 : target phi
o26 = image {2} | 2y+1 2x3y3+x3y2 2x5y2-1 -x5y2-y |
            {1} | -x2  y          x2      0       |
            {7} | 0    -2y-1      -2x2    x2      |
                                     3
o26 : R-module, submodule of R

i27 : isIsomorphism phi
o27 = true

i28 : B = computeFreeBasis K
o28 = {2} | 2x3y3+x3y2 2x5y2-1 |
      {1} | y          x2      |
      {7} | -2y-1      -2x2    |
               3       2
o28 : Matrix R  <--- R

i29 : syz B
o29 = 0
               2
o29 : Matrix R  <--- 0

i30 : image B == K
o30 = true
```

From the *Macaulay2* output, we can see that the native command `mingens` does not produce a free generating set for K, while `computeFreeBasis` produces a set of 2 generators for K with no relations, demonstrating that K is free.

5. Acknowledgements

We thank Amelia Taylor for organizing the 2010 *Macaulay2* workshop at Colorado College[1] where work on the *QuillenSuslin* package began. We would especially like to thank Hirotachi Abo for many useful conversations during the workshop as well as Jason McCullough.

The first author thanks his advisor, Andrew R. Kustin, for many useful discussions during the development of the package.

References

[Fab] Anna Fabianska, *QuillenSuslin Package for Maple*, Available at http://wwwb.math.rwth-aachen.de/QuillenSuslin.

[Fab09] Anna Fabianska, *Algorithmic analysis of presentations of groups and modules*, Ph.D. thesis, RWTH Aachen University, Jan 2009.

[FQ07] Anna Fabianska and Alban Quadrat, *Applications of the Quillen-Suslin Theorem to multidimensional systems theory*, Groöbner bases in control theory and signal processing, Radon Ser. Comput. Appl. Math. **3**, 23–106. MR 2402709 (2009f:13013)

[FG90] Noaï Fitchas and André Galligo, *Nullstellensatz effectif et conjecture de Serre (théorème de Quillen-Suslin) pour le calcul formel*, Math. Nachr. **149** (1990), 231–253. MR 1124807 (92i:12002)

[GS] Daniel R. Grayson and Michael E. Stillman, *Macaulay2, a software system for research in algebraic geometry*, Available at http://www.math.uiuc.edu/Macaulay2/.

[Lam06] T. Y. Lam, *Serre's problem on projective modules*, Springer Monographs in Mathematics, Springer-Verlag, Berlin, 2006. MR 2235330 (2007b:13014)

[LW00] Reinhard C. Laubenbacher and Cynthia J. Woodburn, *A new algorithm for the Quillen-Suslin theorem*, Beiträge Algebra Geom. **41** (2000), no. 1, 23–31. MR 1745576 (2001e:13013)

[LS92] Alessandro Logar and Bernd Sturmfels, *Algorithms for the Quillen-Suslin theorem*, J. Algebra **145** (1992), no. 1, 231–239. MR 1144671 (92k:13006)

[Par04] Hyungju Park, Symbolic computation and signal processing, J. Symbolic Comput. 37 (2004), no. 2, 209–226. MR 2093232 (2005d:94039)

[Qui76] Daniel Quillen, *Projective modules over polynomial rings*, Invent. Math. **36** (1976), 167–171. MR 0427303 (55 #337)

[Rot09] Joseph J. Rotman, *An introduction to homological algebra*, second ed., Universitext, Springer, New York, 2009. MR 2455920 (2009i:18011)

[Ser55] Jean-Pierre Serre, *Faisceaux algébriques cohérents*, Ann. of Math. (2) **61** (1955), 197–278. MR 0068874 (16,953c)

[Sus76] A. A. Suslin, *Projective modules over polynomial rings are free*, Dokl. Akad. Nauk SSSR **229** (1976), no. 5, 1063–1066. MR 0469905 (57 #9685)

[VS76] LN Vaserstein and A. A. Suslin, *Serre's Problem on Projective Modules over Polynomial Rings, and Algebraic K-Theory*, Mathematics of the USSR-Izvestiya **10** (1976), no. 5, 937–1001 (English).

Brett Barwick, Division of Mathematics and Computer Science, University of South Carolina Upstate, Spartanburg, SC 29303

E-mail address: `bbarwick@uscupstate.edu`

Branden Stone, Mathematics Program, Bard College/Bard Prison Initiative, P.O. Box 5000, Annandale-on-Hudson, NY 12504

E-mail address: `bstone@bard.edu`

[1]The *Macaulay2* workshop at Colorado College in Colorado Springs, Colorado was funded by the the National Science Foundation under Grant No. 09-64128 and the National Security Agency under Grant No. H98230-10-1-0218.