Approximate Factorization

Ruppert gave in [6] a criterion for a polynomial to be irreducible. In [3] this criterion was applied to factor polynomials with approximate coefficients, based on the exact algorithms in [2].

1 The Ruppert Matrix

In deriving Ruppert's criterion, we follow the first observation made in [1]. Consider $f = f(x,y)$ and suppose f is reducible and factors as $f = f_1 f_2$. Applying the product rule for derivatives gives

$$\frac{\partial f}{\partial x} = \frac{\partial f_1}{\partial x} f_2 + f_1 \frac{\partial f_2}{\partial x} = g_1 + g_2 \quad \text{and} \quad \frac{\partial f}{\partial y} = \frac{\partial f_1}{\partial y} f_2 + f_1 \frac{\partial f_2}{\partial y} = h_1 + h_2. \tag{1}$$

defining $g_1 = \dfrac{\partial f_1}{\partial x} f_2$, $g_2 = f_1 \dfrac{\partial f_2}{\partial x}$, $h_1 = \dfrac{\partial f_2}{\partial y} f_2$, and $h_2 = f_1 \dfrac{\partial f_2}{\partial y}$. Then we write the derivatives of $\log(f_1)$ as

$$\frac{\partial}{\partial x}(\log(f_1)) = \frac{1}{f_1}\frac{\partial f_1}{\partial x} = \frac{g_1}{f} \quad \text{and} \quad \frac{\partial}{\partial y}(\log(f_1)) = \frac{1}{f_1}\frac{\partial f_1}{\partial y} = \frac{h_1}{f}. \tag{2}$$

For any p with continuous derivatives, the identity $\dfrac{\partial}{\partial x}\left(\dfrac{\partial}{\partial y} p\right) = \dfrac{\partial}{\partial y}\left(\dfrac{\partial}{\partial x} p\right)$ holds and we apply it to $p = \log(f_1)$ and $\log(f_2)$ to find

$$\frac{\partial}{\partial x}\left(\frac{h_1}{f}\right) = \frac{\partial}{\partial y}\left(\frac{g_1}{f}\right) \quad \text{and} \quad \frac{\partial}{\partial x}\left(\frac{h_2}{f}\right) = \frac{\partial}{\partial y}\left(\frac{g_2}{f}\right). \tag{3}$$

Degree bounds on g_1, g_2, h_1, and h_2 will lead to the Ruppert matrix. We denote $\deg_x(f)$ as the degree of f when viewed as a polynomial only in x. Similarly, $\deg_y(f)$ is the degree of f as a polynomial in y only.

The criterion in [6] for f to be irreducible as a partial differential equation is formulated in Theorem 1.1.

Theorem 1.1 *A polynomial $f(x,y) \in \mathbb{C}[x,y]$ is irreducible if and only if*

$$\frac{\partial}{\partial y}\left(\frac{g}{f}\right) = \frac{\partial}{\partial x}\left(\frac{h}{f}\right) \tag{4}$$

has no nonzero solutions for all polynomial $g, h \in \mathbb{C}[x,y]$, with $\deg_x(g) \leq \deg_x(f) - 1$, $\deg_y(g) \leq \deg_y(f)$, and $\deg_x(h) \leq \deg_x(f)$, $\deg_y(h) \leq \deg_y(f) - 2$.

Note that if the condition $\deg_y(h) \leq \deg_y(f) - 2$ on h is changed into $\deg_y(h) \leq \deg_y(f) - 1$, then $g = f_x$ and $h = f_y$ is a solution to (4), regardless whether f is irreducible or not.

Consider for example $f(x,y) = x^2 + y^2 - 1$. Then $g(x,y) = a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{02}y^2 + a_{12}xy^2$ and $h(x,y) = b_{00} + b_{10}x + b_{20}x^2$ are the general forms of the polynomials to satisfy (4). The sequence of commands in Maple generates the Ruppert matrix for f:

```
[> f := x^2 + y^2 - 1;
[> g := sum(sum(a[i,j]*x^i*y^j,i=0..degree(f,x)-1),j=0..degree(f,y));
[> h := sum(sum(b[i,j]*x^i*y^j,i=0..degree(f,x)),j=0..degree(f,y)-2);
[> eq := diff(g/f,y) - diff(h/f,x);
[> nq := normal(eq); p := numer(nq); s := coeffs(p,[x,y]);
[> sys := {seq(s[i]=0,i=1..nops([s]))}; var := indets(sys);
[> R := LinearAlgebra[GenerateMatrix](sys,var)[1];
[> LinearAlgebra[Rank](R);
```

The rank of the matrix `R` equals 9, which equals the number of columns, so f is indeed irreducible.

2 An Open Problem in Symbolic-Numeric Computing

As stated in [1], polynomial factorization is one of the main chapters in computer algebra. In [4], the following problem is stated:

"*Given is a polynomial* $f(x, y) \in \mathbb{Q}[x, y]$ *and* $\epsilon \in \mathbb{Q}$. *Decide in polynomial time in the degree and coefficient size if there is a factorizable* $\overline{f}(x, y) \in \mathbb{C}[x, y]$ *with* $||f - \overline{f}|| \leq \epsilon$, *for a reasonable coefficient vector norm* $||.||$."

In Figure 1 we see plots of an example of [4], once plotted just as given `p` and once after first executing `factor(p,sqrt(2)` and then plotting the factors.

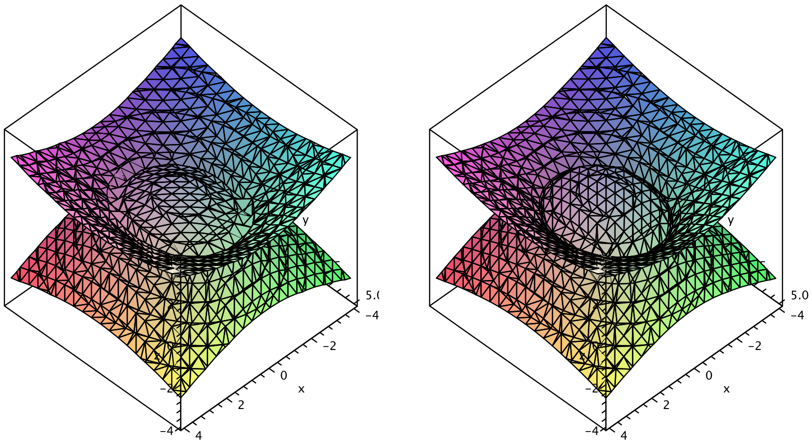

Figure 1: Exported plots of `implicitplot3d` with Maple of $81x^4 + 16y^4 - 648z^4 + 72x^2y^2 - 648x^2 - 288y^2 + 1296$ and $(9x^2 + 4y^2 + 18\sqrt{2}z^2 - 36)(9x^2 + 4y^2 - 18\sqrt{2}z^2 - 36)$.

3 The Kernel of the Ruppert Matrix

Like in [3], we assume that the polynomial f we have to factor is square free, i.e.: $\mathrm{GCD}(f, f_x) = 1$. In [2], the condition on h is relaxed to $\deg_y(h) \leq \deg_y(f) - 1$ and (4) is rewritten into

$$f \cdot \left(\frac{\partial g}{\partial y} - \frac{\partial h}{\partial x} \right) + h \cdot \frac{\partial f}{\partial x} - g \cdot \frac{\partial f}{\partial y} = 0. \tag{5}$$

The relaxation of the condition on h implies that system of linear equations (5) will always have one solution, also if f is irreducible. Moreover, the dimension of the solution space equals the number of irreducible factors of f. Abusing notation, $R(f)$ will still be called the Ruppert matrix, for the matrix resulting of the relaxation on the degree of h.

Once we have a basis of the null space of the Ruppert matrix, how do we then recover the irreducible factors? Solving [1, Exercise 9.2.11], Proposition 3.1 suggests using the GCD.

Proposition 3.1 *Let* $f = f(x, y)$, $f = f_1 f_2 \cdots f_s$, *and* $g_i = \dfrac{\partial f_i}{\partial x} \dfrac{f}{f_i}$, $i = 1, 2, \ldots, s$.

$$v = \sum_{i=1}^{s} \gamma_i g_i, \quad \gamma_i \neq \gamma_j, i \neq j \quad \Rightarrow \quad f_i = \mathrm{GCD}\left(f, v - \gamma_i \frac{\partial f}{\partial x} \right), \quad i = 1, 2, \ldots, s. \tag{6}$$

Proof. To avoid dot dot dots, we assume $s = 3$. The g_i's are defined as $g_1 + g_2 + g_3 = \frac{\partial f}{\partial x}$. Then:

$$\begin{aligned} v - \gamma_1 \frac{\partial f}{\partial x} &= \gamma_1 g_1 + \gamma_2 g_2 + \gamma g_3 - \gamma_1 (g_1 + g_2 + g_3) && (7)\\ &= (\gamma_2 - \gamma_1) g_2 + (\gamma_3 - \gamma_1) g_3 && (8)\\ &= (\gamma_2 - \gamma_1) \frac{\partial f_2}{\partial x} f_1 f_3 + (\gamma_3 - \gamma_1) \frac{\partial f_3}{\partial x} f_1 f_2. && (9) \end{aligned}$$

Because $\gamma_2 \neq \gamma_1$ and $\gamma_3 \neq \gamma_1$ we find

$$\mathrm{GCD}\left(f, v - \gamma_1 \frac{\partial f}{\partial x} \right) = \mathrm{GCD}\left(f_1 f_2 f_3, (\gamma_2 - \gamma_1) \frac{\partial f_2}{\partial x} f_1 f_3 + (\gamma_3 - \gamma_1) \frac{\partial f_3}{\partial x} f_1 f_2 \right) = f_1. \tag{10}$$

The derivations are similar for f_2 and f_3. □

Before we can apply the GCD to the basis elements of the null space we must ensure that these basis elements have the desired form.

Proposition 3.2 *Consider $f = f(x, y)$ and $f = f_1 f_2 \cdots f_s$. Denote by $R(f)$ the Ruppert matrix of the system (5) linear in the coefficient vectors of the polynomials g and h with $\deg(g) \leq (\deg_x(f) - 1, \deg_y(f))$ and $\deg(h) \leq (\deg_x(f), \deg_y(f) - 1)$. Let $\mathbf{u} : R(f)\mathbf{u} = \mathbf{0}$, then $\mathbf{u} = (\mathbf{v}, \mathbf{w})$, where $\mathbf{v}$ and $\mathbf{w}$ are coefficient vectors of the respective polynomials g and h. Identifying the coefficient vector $\mathbf{v}$ with the polynomial $v(x, y)$ we have:*

$$v(x, y) = \sum_{i=1}^{s} \gamma_i g_i(x, y), \quad g_i = \frac{\partial f_i}{\partial x} \frac{f}{f_i}, \ i = 1, 2, \ldots, s, \tag{11}$$

for some constants $\gamma_i \in \mathbb{C}$.

Proof. Assuming f is monic, we write f as a function of y, expressing the values for the x-coordinates of f as $x_i(y)$, $i = 1, 2, \ldots, d$, where $d = \deg_x(f)$:

$$f(x(y), y) = \prod_{i=1}^{d} (x - x_i(y)). \tag{12}$$

Since $\deg_x(v) < \deg_x(f)$, we have a partial fraction decomposition

$$\frac{v}{f} = \sum_{i=1}^{d} \frac{a_i(y)}{x - x_i(y)}, \quad a_i(y) = \frac{v(x_i(y), y)}{\displaystyle\prod_{j \neq i} (x - x_i(y))} = \frac{v(x_i(y), y)}{\frac{\partial f}{\partial x}(x_i(y), y)}. \tag{13}$$

We obtain the expression for $a_i(y)$ by equating numerators in the partial fraction decomposition identity for v/f. For the polynomial $w(x, y)$ with coefficient vector $\mathbf{w}$, corresponding to the coefficient vector $\mathbf{v}$ of v, we also set up a partial fraction decomposition:

$$\frac{w}{f} = \sum_{i=1}^{d} \frac{b_i(y)}{x - x_i(y)} + b_0, \quad b_0 \in \mathbb{C}. \tag{14}$$

Because $\mathbf{u} = (\mathbf{v}, \mathbf{w})$ belongs to the kernel of $R(f)$: $\frac{\partial}{\partial y}\left(\frac{v}{f}\right) = \frac{\partial}{\partial x}\left(\frac{w}{f}\right)$. Applying this property to the partial fraction decompositions in (13) and (14):

$$\begin{aligned} \frac{\partial}{\partial y}\left(\frac{w}{f}\right) &= \sum_{i=1}^{d} \frac{-b_i^2}{(x - x_i(y))^2} && (15)\\ \frac{\partial}{\partial x}\left(\frac{v}{f}\right) &= \sum_{i=1}^{d} \frac{1}{x - x_i(y)} \frac{\partial a_i}{\partial y} + \sum_{i=1}^{d} \frac{a_i}{(x - x_i(y))^2} \left(-\frac{\partial x_i}{\partial y} \right). && (16) \end{aligned}$$

So we find that $\frac{\partial}{\partial y}\left(\frac{v}{f}\right) = \frac{\partial}{\partial x}\left(\frac{w}{f}\right)$ implies $\frac{\partial a_i}{\partial y} = 0$.

The constant coefficients a_i belonging to the same factor f_k of f are all conjugated and are all equal, say to γ_k. So we may write

$$\frac{v}{f} = \sum_{k=1}^{s} \frac{\gamma_k}{\prod_j (x - x_j(y))} = \sum_{k=1}^{s} \gamma_k \frac{\partial f_k}{\partial x}\frac{1}{f_k}. \tag{17}$$

Therefore $v = \sum_{k=1}^{s} \gamma_k \frac{\partial f_k}{\partial x}\frac{f}{f_k} = \sum_{k=1}^{s} \gamma_k g_k$. □

Although we now know that a basis of the kernel of $R(f)$ gives linear combinations of the g_i's, to apply Proposition 3.1, we need the actual g_i's itself. Our third proposition brings us there.

Proposition 3.3 *Let the matrix* $V = [\mathbf{v}_1\mathbf{v}_2\cdots\mathbf{v}_s]$ *collect the components of the basis vectors of the kernel of the Ruppert matrix* $R(f)$, *i.e.* $R(f)\mathbf{u} = \mathbf{0}$, $\mathbf{u} = (\mathbf{v}, \mathbf{w})$, $\mathbf{v}$ *contains the coefficient vectors of the polynomials* g *of Ruppert's criterion. For any* $\mathbf{v}$ *in the span of* V, *there is a unique matrix* $A \in \mathbb{C}^{s\times s}$*:*

$$\mathbf{v}\mathbf{v}_i = \sum_{j=1}^{s} a_{ij}\mathbf{v}_j \frac{\partial f}{\partial x} \mod f \quad \text{in } \mathbb{C}(y)[x]. \tag{18}$$

Moreover:

$$f = \prod_{\substack{\lambda \in \mathbb{C} \\ \det(A - \lambda I) = 0}} \mathrm{GCD}\left(f, v - \lambda\frac{\partial f}{\partial x}\right), \tag{19}$$

i.e.: the i*th irreducible factor of* f *is* $f_i = \mathrm{GCD}\left(f, v - \lambda_i\frac{\partial f}{\partial x}\right)$, *where* λ_i *the* i*th eigenvalue of* A.

Proof. Because $\mathbf{v} = \gamma_1 g_1 + \gamma_2 g_2 + \cdots + \gamma_s g_s$, with $g_i = \frac{f}{f_i}\frac{\partial f}{\partial x}$, the second statement of the proposition follows immediately if $\gamma_i = \lambda_i$, as f_i divides $\mathbf{v} - \lambda_i\frac{\partial f}{\partial x}$.

Since V is a basis for the null space of the Ruppert matrix, there exists an s-by-s matrix B such that

$$\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_s \end{bmatrix} = B \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_s \end{bmatrix}. \tag{20}$$

We have that $g_i g_j = \left(\frac{\partial f_i}{\partial x}\prod_{\substack{k=1\\ \neq i}}^{s} f_k\right)\left(\frac{\partial f_j}{\partial x}\prod_{\substack{k=1\\ \neq j}}^{s} f_k\right)$ is a multiple of f for $i \neq j$, so $g_i g_j \equiv 0 \mod f$. Then we can write

$$\mathbf{v}\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_s \end{bmatrix} \equiv B \begin{bmatrix} \mathbf{v}g_1 \\ \mathbf{v}g_2 \\ \vdots \\ \mathbf{v}g_s \end{bmatrix} \equiv B \begin{bmatrix} \lambda_1 g_1^2 \\ \lambda_2 g_2^2 \\ \vdots \\ \lambda_s g_s^s \end{bmatrix} \equiv B \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_s \end{bmatrix} \begin{bmatrix} g_1^2 \\ g_2^2 \\ \vdots \\ g_s^2 \end{bmatrix} \mod f. \tag{21}$$

The multiplication of $\dfrac{\partial f}{\partial x}$ with V leads to

$$\frac{\partial f}{\partial x} \left[\begin{array}{c} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_s \end{array} \right] \equiv B \left[\begin{array}{c} \frac{\partial f}{\partial x} g_1 \\ \frac{\partial f}{\partial x} g_2 \\ \vdots \\ \frac{\partial f}{\partial x} g_s \end{array} \right] \equiv B \left[\begin{array}{c} g_1^2 \\ g_2^2 \\ \vdots \\ g_s^2 \end{array} \right] \mod f, \quad \text{as } \frac{\partial f}{\partial x} = \sum_{i=1}^s g_i. \tag{22}$$

So we substitute

$$\left[\begin{array}{c} g_1^2 \\ g_2^2 \\ \vdots \\ g_s^2 \end{array} \right] \equiv B^{-1} \frac{\partial f}{\partial x} \left[\begin{array}{c} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_s \end{array} \right] \mod f \tag{23}$$

into (21) and find that the matrix

$$A = B \left[\begin{array}{cccc} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_s \end{array} \right] B^{-1} \tag{24}$$

has eigenvalues λ_i, $i = 1, 2, \ldots, s$. □

The three propositions lead to Gao's theorem [1] and his algorithm to factor polynomials, described in [2].

4 SVD and Approximate GCD

In [3], the authors propose to use the singular value decomposition (SVD) to compute a basis for the null space of the linear system defined by (5). This basis is formed by the last r right singular vectors of the coefficient matrix of the linear system. The rank r of the null space occurs at the biggest gap $\max_i(\sigma_{i+1}/\sigma_i)$ in the singular values σ_i.

Once a basis for the null space is computed, a random combination p of the polynomials corresponding to the basis vectors is taken. Fixing $y = \alpha \in \mathbb{C}$, where α is random, the roots λ_i of p determine the factors of f: $f_i = \mathrm{GCD}(f, v - \lambda_i f_x)$, for $i = 1, 2, \ldots, r$, for v a basis vector in the null space of the Ruppert matrix. The algorithm ends with a computation of the backward error. In [5], Gauss-Newton is applied to improve the accuracy of the computed factors.

The algorithm below is an adaptation from [3].

Algorithm 4.1 (Approximate Bivariate Factorization)

```
Input: $f \in \mathbb{C}[x,y]$, $\mathrm{GCD}(f, \frac{\partial f}{\partial x}) = 1$, $f$ has no approximate factors in $\mathbb{C}[y]$;
        $S \subset \mathbb{C}$ and $\#S \geq \deg_x(f) \times \deg_y(f)$.
Output: list of approximate factors of $f$.

Stage 1: form the Ruppert matrix $R(f)$;
         find the last $\deg(f)+1$ singular values $\sigma_i$ of $R(f)$, $\sigma_n \geq \sigma_{n-1} \geq \cdots \geq \sigma_2 \geq \sigma_1$;
         let $s$ be the index so $\sigma_{s+1}/\sigma_s$ is maximal;
         if $s = 1$, then return $f$;
         form a basis $\mathbf{v}_1$, $\mathbf{v}_1$, ..., $\mathbf{v}_s$ from the last $s$ right singular vectors of $R(f)$;
```

Stage 2: $\mathbf{v} := \displaystyle\sum_{s_i \in S} s_i \mathbf{v}_i$, with coefficients s_i selected uniformly and independently;

for $y = \alpha$, compute a_{ij} that minimize $\left\| \text{remainder} \left(\mathbf{v}\mathbf{v}_i - \displaystyle\sum_{j=1}^{s} a_{ij} \mathbf{v}_j \frac{\partial f}{\partial x}, f \right) \right\|_2$;

compute the eigenvalues λ_i of $A = [a_{ij}]$;

Stage 3: $f_i := \mathrm{GCD}\left(f, \mathbf{v} - \lambda_i \dfrac{\partial f}{\partial x} \right)$, for $i = 1, 2, \ldots, s$, where GCD is an approximate GCD.

We end with some comments on the algorithm. The algorithm in [3] chooses another selection of the values s_i of S is the computed eigenvalues are clustered. The matrix A is formed reducing $\mathbf{v}\mathbf{v}_i$ and $g_j \dfrac{\partial f}{\partial x}$ with respect to $y = \alpha$ via a least squares problem. The algorithm in [3] ends with a backward error computation and iterative refinement of the approximate factors.

5 Exercises

1. Find a general formula for the size of the Ruppert matrix, in terms of the degrees $\deg_x(f)$ and $\deg_y(f)$.

2. Show that for $f = f(x,y)$, $f = f_1 f_2$, $g_1 = \frac{\partial f_1}{\partial x} f_2$, $g_2 = f_1 \frac{\partial f_2}{\partial x}$, $h_1 = \frac{\partial f_2}{\partial y} f_2$, and $h_2 = f_1 \frac{\partial f_2}{\partial y}$:
$$f \left(\frac{\partial g_1}{\partial y} - \frac{\partial h_1}{\partial x} \right) + h_1 \frac{\partial f}{\partial x} - g_1 \frac{\partial f}{\partial y} = 0. \qquad (25)$$

3. Download the Maple code at
`http://www4.ncsu.edu/~kaltofen/software/appfac/issac04_mws/multifac_1.3.mpl`
and use it to factor (example from [2]):
$$f(x,y) = 9 + 23y^2 + 13yx^2 + 6y + 7y^3 + 13y^2x^2 + x^4 + 6yx^4 + x^6. \qquad (26)$$

4. Download `ApaTools` available via [7] and use it to factor the example (26) from the previous exercise.

   Instead of (or in addition to) `ApaTools`, you may also consider NAClab at `www.neiu.edu/~naclab` to solve the exercise.

5. Consider f in (26), but now add some random errors to the coefficients, of magnitude 10^{-k}, for k ranging from 1 to 14. For $k = 1$, f is irreducible, while for $k = 14$, the numerical algorithm should return the same factorization as in the exact case. For which k is f no longer irreducible?

References

[1] G. Chèze and A. Galligo. Four lectures on polynomial absolute factorization. In A. Dickenstein and I.Z. Emiris, editors, *Solving Polynomial Equations. Foundations, Algorithms and Applications*, volume 14 of *Algorithms and Computation in Mathematics*, pages 339–394. Springer–Verlag, 2005.

[2] S. Gao. Factoring multivariate polynomials via partial differential equations. *Mathematics of Computation*, 72(242):801–822, 2002.

[3] S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials via differential equations. In J. Gutierrez, editor, *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation (ISSAC 2004)*, pages 167–174. ACM, 2004.

[4] E. Kaltofen. Challenges of symbolic computation: my favorite open problems. *J. Symbolic Computation*, 29(6):891–919, 2000.

[5] E. Kaltofen, J.P May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials using singular value decomposition. *Journal of Symbolic Computation*, 43(5):359–376, 2008.

[6] W.M. Ruppert. Reducibility of polynomials $f(x, y)$ modulo p. *Journal of Number Theory*, 77(1):62–70, 1999.

[7] Z. Zeng. ApaTools: a software toolbox for approximate polynomial algebra. In M.E. Stillman, N. Takayama, and J. Verschelde, editors, *Software for Algebraic Geometry*, volume 148 of *The IMA Volumes in Mathematics and Its Applications*, pages 149–167. Springer-Verlag, 2008.