

A New Algorithm for Sparse Interpolation of Multivariate Polynomials

Annie Cuyt* and Wen-shin Lee

*Department of Mathematics and Computer Science, University of Antwerp
Middelheimlaan 1, 2020 Antwerpen, Belgium*

Abstract

To reconstruct a black box multivariate sparse polynomial from its floating point evaluations, the existing algorithms need to know upper bounds for both the number of terms in the polynomial and the partial degree in each of the variables. Here we present a new technique, based on Rutishauser's qd -algorithm, in which we overcome both drawbacks.

Key words: black box polynomial, symbolic-numeric sparse interpolation, qd -algorithm, generalized eigenvalue, Hadamard polynomial, early termination.

1 Introduction

The reconstruction of a multivariate polynomial

$$p(x_1, \dots, x_n) = \sum_{(j_1, \dots, j_n) \in J} c_{j_1 \dots j_n} x_1^{j_1} \cdots x_n^{j_n}, \quad J \subset \mathbb{N}^n$$

from some function evaluations is easy if the support J is known. It suffices to have as many function evaluations as the cardinality of J and to write down a linear system of interpolation conditions. In this paper we focus on the situation where neither J nor its cardinality is known, in other words neither the number of non-zero monomials in $p(x_1, \dots, x_n)$ nor their exponents (j_1, \dots, j_n) is known, and the evaluations of $p(x_1, \dots, x_n)$ are performed in

* Corresponding author

Email addresses: annie.cuyt@ua.ac.be (Annie Cuyt),
wen-shin.lee@ua.ac.be (Wen-shin Lee).

URLs: <http://www.win.ua.ac.be/~cant> (Annie Cuyt),
<http://www.wen-shin.com> (Wen-shin Lee).

floating point arithmetic. We remark that in exact arithmetic the number of non-zero terms in the polynomial can be detected using a probabilistic strategy called early termination [9; 8], but this technique is not applicable here.

A number of techniques are available in a floating point context. Our presentation order is at the same time chronological and increasing in generality. The first sparse interpolation algorithm was given in 1979 by Zippel [11]. We depart from a floating point technique [4] based on the 1988 algorithm by Ben-Or and Tiwari [2; 7], which assume that upper bounds p_k for the partial degrees of p in each of the variables x_k and (an estimate of) the cardinality of the support are known. A reformulation of the problem as a generalized eigenvalue problem by Golub, Milanfar and Varah in 1999 [5], under the same assumptions, eliminates the computation of some intermediate values and offers a stable numerical algorithm. We present an alternative algorithm which does not require the knowledge of the cardinality of the support J nor of a bounding box $J \subset [0, p_1] \times \cdots \times [0, p_n] \cap \mathbb{N}^n$.

Let us explain the basic theory underlying all algorithms in [2; 7; 4]. We denote the evaluation of the black box polynomial $p(x_1, \dots, x_n)$ at the point $(x_1, \dots, x_n) = (\xi_1^s, \dots, \xi_n^s)$ by

$$\pi_s = p(\xi_1^s, \dots, \xi_n^s), \quad s = 0, 1, \dots$$

Note that the evaluation points are s -th powers of some (suitably chosen) vectors (ξ_1, \dots, ξ_n) . Let us enumerate the t multi-indices in J as

$$\left(j_1^{(i)}, \dots, j_n^{(i)} \right), \quad i = 1, \dots, t$$

and introduce for $i = 1, \dots, t$ the abbreviate notations $\beta_i = \xi_1^{j_1^{(i)}} \cdots \xi_n^{j_n^{(i)}}$, $c_i = c_{j_1^{(i)} \dots j_n^{(i)}}$. Let us assume that all β_i are distinct. We have

$$\pi_s = \sum_{i=1}^t c_i \beta_i^s.$$

Now set

$$z^t + a_{t-1}z^{t-1} + \cdots + a_0 = \prod_{i=1}^t (z - \beta_i).$$

Since the β_i are the zeros of this monic polynomial, we find

$$\begin{aligned} 0 &= \sum_{i=1}^t c_i \beta_i^s \left(\beta_i^t + a_{t-1} \beta_i^{t-1} + \cdots + a_0 \right) \\ &= \sum_{i=1}^t c_i \beta_i^{s+t} + \sum_{j=0}^{t-1} a_j \sum_{i=1}^t \left(c_i \beta_i^{s+j} \right) = \pi_{t+s} + \sum_{j=0}^{t-1} a_j \pi_{j+s}. \end{aligned}$$

Hence the sequence of polynomial evaluations π_s at the s -th powers is linearly generated. Since the β_i are distinct, one can prove that the monic polynomial $z^t + a_{t-1}z^{t-1} + \dots + a_0$ is the polynomial of minimal degree with this property.

2 A numeric Ben-Or/Tiwari algorithm

In a floating point context, a suitable choice for the vectors (ξ_1, \dots, ξ_n) means a choice that keeps the involved linear systems well-conditioned. Since s -th powers of these vectors are taken, we place them on the unit circle to avoid a growth of magnitude [4]. Let the positive integers p_k for $k = 1, \dots, n$ be mutually prime and bound the partial degree of p in the variable x_k , hence $p_k > \partial_{x_k} p$. Set

$$m = p_1 \cdots p_n, \quad \omega = \exp(2\pi i/m), \quad \omega_k = \omega^{m/p_k}, \quad k = 1, \dots, n. \quad (1)$$

Let the cardinality t of J be given (or an upper bound estimated). Evaluate

$$\pi_s = p(\omega_1^s, \dots, \omega_n^s), \quad 0 \leq s \leq 2t - 1$$

at the roots of unity and solve for the coefficients of the monic polynomial $z^t + a_{t-1}z^{t-1} + \dots + a_0$ from the Hankel system

$$\begin{bmatrix} \pi_0 & \pi_1 & \dots & \pi_{t-1} \\ \pi_1 & \pi_2 & \dots & \pi_t \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{t-1} & \pi_t & \dots & \pi_{2t-2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{bmatrix} = - \begin{bmatrix} \pi_t \\ \pi_{t+1} \\ \vdots \\ \pi_{2t-1} \end{bmatrix}.$$

The algorithm is based on the fact that the sequence of evaluations π_s is linearly generated by

$$\pi_s + a_{t-1}\pi_{s-1} + \dots + a_0\pi_{s-t} = 0.$$

From [2], it is known that the t roots of this monic polynomial are of the form $\omega^{j^{(i)}}$ where

$$j^{(i)} = j_1^{(i)} \frac{m}{p_1} + \dots + j_n^{(i)} \frac{m}{p_n}, \quad (j_1^{(i)}, \dots, j_n^{(i)}) \in J, \quad i = 1, \dots, t.$$

Consequently the values $j^{(i)}$ can be retrieved from the roots $\omega^{j^{(i)}}$ and the individual $j_k^{(i)}$ can be obtained from $j^{(i)}$ through a reverse application of the Chinese remainder theorem [4]. Note that the values $j^{(i)}$ are integers, which simplifies their computation since rounding errors are present in $\omega^{j^{(i)}}$. To know

the polynomial $p(x_1, \dots, x_n)$ it suffices to determine the coefficients $c_{j_1 \dots j_n}$ from the solution of a classical Vandermonde system.

In exact arithmetic the black box polynomial p is evaluated at (ξ_1, \dots, ξ_n) with the ξ_k pairwise relatively prime integer numbers. When picking them randomly and computing the so-called discrepancy Δ_s where

$$\Delta_{s+1} = \pi_s + a_{t-1}\pi_{s-1} + \dots + a_0\pi_{s-t}$$

the guess for the cardinality t of J can be updated with high probability when the next element π_{2t} does not fit the current linear recursion [9]. In a floating-point context however, this strategy does not work.

3 A generalized eigenvalue algorithm

Now let us denote the Hankel matrices

$$H_t^{(s)} = \begin{bmatrix} \pi_s & \cdots & \pi_{s+t-1} \\ \vdots & \ddots & \pi_{s+t} \\ & & \vdots \\ \pi_{s+t-1} & \cdots & \pi_{s+2t-2} \end{bmatrix}, \quad s \geq 0, t \geq 1,$$

the polynomials

$$H_t^{(s)}(z) = \begin{vmatrix} \pi_s & \cdots & \pi_{s+t-1} & \pi_{s+t} \\ \vdots & \ddots & & \\ & & \vdots & \vdots \\ \pi_{s+t-1} & \cdots & & \pi_{s+2t-1} \\ 1 & \dots & z^{t-1} & z^t \end{vmatrix}, \quad H_0^{(s)}(z) = 1$$

and the linear functional γ that associates

$$\gamma(z^s) = \pi_s.$$

Then the monic polynomial

$$z^t + a_{t-1}z^{t-1} + \dots + a_0 = \frac{H_t^{(0)}(z)}{\det H_t^{(0)}}. \quad (2)$$

It is a formally orthogonal polynomial satisfying [3, pp. 40–41]

$$\gamma \left(z^i H_t^{(0)}(z) / \det H_t^{(0)} \right) = 0, \quad i = 0, \dots, t-1$$

and is called a Hadamard polynomial [6, pp. 625]. More generally one can define the monic Hadamard polynomials

$$p_0^{(s)}(z) = 1, \quad p_t^{(s)}(z) = \frac{H_t^{(s)}(z)}{\det H_t^{(s)}}, \quad s, t = 1, 2, \dots$$

It is proved in [5] that the t roots of the monic polynomial (2) can be obtained by solving the generalized eigenvalue problem

$$H_t^{(1)}v = zH_t^{(0)}v, \quad v \in \mathbb{C}^t.$$

Hence the explicit computation of the coefficients a_0, \dots, a_{t-1} and the corresponding root finding can be skipped. The roots can be obtained directly from the generalized eigenvalue problem. The sequel remains as above, deducing the multi-indices $(j_1^{(i)}, \dots, j_n^{(i)})$ in the support J from the polynomial roots and the coefficients $c_{j_1 \dots j_n}$ from an interpolation problem. When a wrong estimate for t is made, one can verify a posteriori whether the evaluations of the reconstructed polynomial $p(x_1, \dots, x_n)$ match some new function values obtained from the black box probe.

In [6, p. 635] it is also pointed out that the roots of $p_t^{(s)}(z)$ are the eigenvalues of a particular tridiagonal matrix, a result that we make use of in the next section.

4 Sparse interpolation using the qd-algorithm

With the sequence $\{\pi_s\}_{s \in \mathbb{N}}$ we can also set up the qd -scheme, where subscripts denote columns and superscripts denote downward sloping diagonals [6]. Its initialization is given by

$$\begin{aligned} e_0^{(s)} &= 0, & s &= 1, 2, \dots \\ q_1^{(s)} &= \frac{\pi_{s+1}}{\pi_s}, & s &= 0, 1, \dots \end{aligned}$$

and the rhombus rules for continuation of the scheme by

$$\begin{aligned} e_u^{(s)} &= q_u^{(s+1)} - q_u^{(s)} + e_{u-1}^{(s+1)}, & u &= 1, 2, \dots, & s &= 0, 1, \dots \\ q_{u+1}^{(s)} &= \frac{e_u^{(s+1)}}{e_u^{(s)}} q_u^{(s+1)}, & u &= 1, 2, \dots, & s &= 0, 1, \dots \end{aligned} \quad (3)$$

In its more stable progressive form the same qd -scheme is initialized with

$$e_0^{(s)} = 0, \quad s = 1, 2, \dots$$

$$q_u^{(0)} = \frac{\det H_{u-1}^{(0)} \det H_u^{(1)}}{\det H_u^{(0)} \det H_{u-1}^{(1)}}, \quad e_u^{(0)} = \frac{\det H_{u+1}^{(0)} \det H_{u-1}^{(1)}}{\det H_u^{(0)} \det H_u^{(1)}}, \quad u = 1, 2, \dots$$

and continued with

$$q_u^{(s+1)} = e_u^{(s)} - e_{u-1}^{(s+1)} + q_u^{(s)}, \quad s = 0, 1, \dots$$

$$e_u^{(s+1)} = \frac{q_{u+1}^{(s)}}{q_u^{(s+1)}} e_u^{(s)}, \quad s = 0, 1, \dots$$

In [6, pp. 634–636] it is subsequently shown that

$$p_t^{(s)}(z) = \det(zI - A_t^{(s)})$$

where $A_t^{(s)}$ denotes the matrix

$$A_t^{(s)} = \begin{bmatrix} q_1^{(s)} + e_0^{(s)} & q_1^{(s)} e_1^{(s)} & & & 0 \\ 1 & q_2^{(s)} + e_1^{(s)} & q_2^{(s)} e_2^{(s)} & & \\ & \ddots & \ddots & \ddots & \\ & & & 1 & q_{t-1}^{(s)} + e_{t-2}^{(s)} & q_{t-1}^{(s)} e_{t-1}^{(s)} \\ 0 & & & & 1 & q_t^{(s)} + e_{t-1}^{(s)} \end{bmatrix}.$$

Hence the zeros of the Hadamard polynomials are the eigenvalues of the matrix $A_t^{(s)}$, or equivalently of the matrix $B_t^{(s)}$ where

$$B_t^{(s)} = \begin{bmatrix} q_1^{(s)} + e_0^{(s)} & -q_1^{(s)} & & & 0 \\ -e_1^{(s)} & q_2^{(s)} + e_1^{(s)} & -q_2^{(s)} & & \\ & \ddots & \ddots & \ddots & \\ & & & -e_{t-2}^{(s)} & q_{t-1}^{(s)} + e_{t-2}^{(s)} & -q_{t-1}^{(s)} \\ 0 & & & & -e_{t-1}^{(s)} & q_t^{(s)} + e_{t-1}^{(s)} \end{bmatrix}.$$

In [10, p. 467] we read that the inverses of these eigenvalues are the poles of the function

$$g^{(s)}(z) = \frac{z}{1 + \frac{-q_1^{(s)}z}{1 + \frac{-e_1^{(s)}z}{1 + \dots + \frac{-q_t^{(s)}z}{1}}}}.$$

And in [6, p. 626] it is proved that for fixed t , the function $g^{(s)}(z)$ is independent of the superscript s . Hence, for each s , the polynomials $p_t^{(s)}(z)$ have the same roots which can be determined as the inverses of the poles of $g^{(s)}(z)$.

The next theorem [6, Theorems 7.6a-b, 7.7d-f] is a combination of all these facts. It tells us that the qd -algorithm, when initialized with the column $e_0^{(s)}$ and the diagonal consisting of $q_u^{(0)}$ and $e_u^{(0)}$, can be an ingenious way to obtain the zeros of the Hadamard polynomials and detect t at the same time. None of the previously discussed methods (Sections 2 and 3) can deliver the value of t , the number of non-zero terms in the polynomial $p(x_1, \dots, x_n)$.

Theorem 4.1 *Let the roots z_j of $H_t^{(0)}(z)/\det H_t^{(0)}$ be numbered such that*

$$|z_1| \geq |z_2| \geq \dots \geq |z_t| > 0 = |z_{t+1}|$$

each root occurring as many times in this sequence as indicated by its multiplicity. Then the qd -scheme has the following properties:

(a) *for each u with $0 < u \leq t$ and $|z_u| > |z_{u+1}|$, it holds that*

$$\lim_{s \rightarrow \infty} e_u^{(s)} = 0;$$

(b) *for each u with $0 < u \leq t$ and $|z_{u-1}| > |z_u| > |z_{u+1}|$, it holds that*

$$\lim_{s \rightarrow \infty} q_u^{(s)} = z_u;$$

(c) *for each u and $\ell > 1$ such that $0 < u < u + \ell \leq t$ and $|z_{u-1}| > |z_u| = \dots = |z_{u+\ell-1}| > |z_{u+\ell}|$, it holds that for the polynomials $\rho_i^{(s)}$ defined by*

$$\begin{aligned} \rho_0^{(s)}(z) &= 1, \\ \rho_{j+1}^{(s)}(z) &= z\rho_j^{(s+1)}(z) - q_{u+j+1}^{(s)}\rho_j^{(s)}(z), \quad s \geq 0, \quad j = 0, 1, \dots, \ell - 1, \end{aligned}$$

there exists a subsequence that converges to

$$(z - z_u) \dots (z - z_{u+\ell-1});$$

(d) *for $u = t$ we have*

$$e_t^{(s)} = 0, \quad s > 0.$$

Theorem 4.1 gives sufficient conditions to guarantee that the qd -table is divided into subtables by e -columns that tend to zero. Any q -column corresponding to a simple zero of isolated modulus is flanked by such e -columns and converges to the corresponding zero. If a subtable contains $\ell > 1$ columns of q -values, the presence of ℓ zeros of equal modulus is indicated. While the e -values in the columns $1, \dots, t-1$ can be small, the values $e_t^{(s)}$ are actually zero (up to rounding errors). This difference is easily distinguishable and allows us to detect the value of t . Moreover, in [1] a combination of the qd -algorithm with a deflation technique leads to necessary conditions to come to the same conclusion. Hence t does not need to be found by trial and error anymore.

There remains the problem of choosing the evaluation points (ξ_1, \dots, ξ_n) suitably. When taken equimodular, such as in Section 2 where $\xi_k = \omega_k$, then we need case (c) of Theorem 4.1. The advantage is that none of the intermediate columns $e_u^{(s)}, 1 \leq u \leq t-1$, converges to zero and hence the continuation rule (3) need not be unstable (no small values in the denominator of $q_{u+1}^{(s)}$). But an upper bound p_k for the partial degree in each variable x_k , as in (1), is required in the input. In addition, one has to solve a polynomial root finding problem.

When taking all $\xi_k, 1 \leq \xi_k \leq n$, relatively prime or equal to the reciprocals of relatively prime numbers, then their powers $(\xi_1^s, \dots, \xi_n^s)$ are different in modulus. The roots of $H_t^{(0)}$ are all simple and also different in modulus and (a) and (b) in Theorem 1 apply. Here each pole is clearly delivered individually as the limit $\lim_{s \rightarrow \infty} q_u^{(s)}, 1 \leq u \leq t$, which is an advantage. But each q -column is now flanked by an e -column which converges to zero, which may be considered as a slight disadvantage. Anyway, the user of the algorithm has the freedom of choice for the points (ξ_1, \dots, ξ_n) .

The second choice for (ξ_1, \dots, ξ_n) can provide the upper bounds p_k needed in the first choice. Also a wrong guess for p_k can easily be invalidated. In both cases the retrieved zeros z_u have to be rounded. In case (c) we round to an integer power of ω given by (1). In case (a) and (b), we round to an integer (or its reciprocal), which is a product of the chosen ξ_k . The multivariate exponent can be recovered as described. Both choices are illustrated in Section 5. In [1] a breakdown free version of the qd -algorithm is described. As mentioned earlier, it combines the continuation rules with a deflation technique.

5 Numerical illustration

We illustrate the above with the reconstruction of

$$p(x, y, z) = \pi x^5 y^7 z - e y z^{11} - \frac{\sqrt{2}}{10} x^9 z^3 + 100 z^3.$$

The floating point version of the Ben-Or/Tiwari algorithm [4] requires estimates for t and a degree upper bound p_k in each variable as input. The purpose of this paper is to show that alternatives exist where none is required. Therefore we illustrate what happens when a wrong guess for t is made.

Let us choose $p_1 = 17, p_2 = 11, p_3 = 13$ and hence

$$\omega_1 = \exp(2\pi i/17), \quad \omega_2 = \exp(2\pi i/11), \quad \omega_3 = \exp(2\pi i/13).$$

When we guess $t = 3$ then the four terms in $p(x, y, z)$ with real coefficients collapse into

$$-(1.4656 + 0.6645i)x^{13}y^2z^{12} + (1.7008 + 0.9511i)x^2yz^{10} + (100.04 - 0.2866i)z^3.$$

Of course, additional a posteriori evaluations of this reconstruction for $p(x, y, z)$ quickly invalidate the model.

Let us now run the qd -algorithm, for a start with the same data. The magnitude of the top few values in the first three e -columns varies between 10^{-2} and 10 while it drops to the order of 10^{-10} in the fourth e -column. This is a clear indication that $t = 4$. It is not difficult to find the four equimodular roots which round to (as required by the theory) $\omega^{2278}, \omega^{1848}, \omega^{561}, \omega^{18}$ for $\omega = \exp 2\pi i/m$ with $m = 17 \cdot 11 \cdot 13$. To recover the multivariate exponents from the integer exponents of ω we write modulo m :

$$\begin{aligned} 1848 &= 9(m/p_1) + 3(m/p_3), \\ 561 &= 3(m/p_3), \\ 2278 &= 1(m/p_2) + 11(m/p_3), \\ 18 &= 5(m/p_1) + 7(m/p_2) + 1(m/p_3). \end{aligned}$$

The corresponding coefficients are found to be

$$p(x, y, z) \approx 3.1415 x^5 y^7 z - 2.7182 yz^{11} - 0.1414 x^9 z^3 + 99.999 z^3.$$

Here we have neglected any imaginary parts in the coefficients of the order of 10^{-13} and smaller.

We conclude and show that essentially neither t nor p_k is required in the input. When evaluating the black box polynomial $p(x, y, z)$ at the non-equidistant s -th powers of $\xi_1 = 1/3, \xi_2 = 1/5, \xi_3 = 1/2$, in which 3, 5, 2 are pairwise relatively prime, then the magnitude of the values in the first three e -columns drops from 10^{-2} to machine precision, while all values in the fourth e -column are of the order of machine precision. Hence again clearly $t = 4$ and now the

multi-indices in the support J are directly obtained from the q -values:

$$\begin{aligned} 1/q_1^{(s)} &\rightarrow 8 = 2^3, \\ 1/q_2^{(s)} &\rightarrow 10240 = 2^{11} 5^1, \\ 1/q_3^{(s)} &\rightarrow 157464 = 3^9 2^3, \\ 1/q_4^{(s)} &\rightarrow 37968750 = 3^5 5^7 2^1. \end{aligned}$$

In this way the non-zero terms $z^3, yz^{11}, x^9z^3, x^5y^7z$ are retrieved and the reconstruction of the coefficients in $p(x, y, z)$ is as above.

References

- [1] H. Allouche, A. Cuyt, Reliable pole detection using a deflated qd-algorithm: When Bernoulli, Hadamard and Rutishauser cooperate, *Numerische Mathematik* (in press).
- [2] M. Ben-Or, P. Tiwari, A deterministic algorithm for sparse multivariate polynomial interpolation, in: *Proc. Twentieth Annual ACM Symp. Theory Comput. STOC'88*, ACM, 1988.
- [3] C. Brezinski, *Padé Type Approximation and General Orthogonal Polynomials*, in: *ISNM*, vol. 50, Birkhauser Verlag, Basel, 1980.
- [4] M. Giesbrecht, G. Labahn, W.-s. Lee, Symbolic-numeric sparse interpolation of multivariate polynomials [extended abstract], in: *Proc. 2006 Internat. Symp. Symbolic Algebraic Comput. ISSAC 2006*, ACM, 2006.
- [5] G. Golub, P. Milanfar, J. Varah, A stable numerical method for inverting shape from moments, *SIAM J. Sci. Comput.* 21 (1999) 1222–1243.
- [6] P. Henrici, *Applied and Computational Complex Analysis I*, John Wiley, New York, 1974.
- [7] E. Kaltofen, Lakshman Yagati, Improved sparse multivariate polynomial interpolation algorithms, in: *Proc. Internat. Symp. ISSAC'88 Symbolic Algebraic Comput*, in: *LNCS*, vol. 358, Springer, 1988.
- [8] E. Kaltofen, W.-s. Lee, Early termination in sparse interpolation algorithms, *J. Symbolic Comput.* 36 (3-4) (2003) 365–400.
- [9] E. Kaltofen, W.-s. Lee, A.A. Lobo, Early termination in Ben-Or/Tiwari sparse interpolation and a hybrid of Zippel's algorithm, in: *Proc. 2000 Internat. Symp. Symbolic Algebraic Comput., ISSAC 2000*, ACM, 2000.
- [10] H. Rutishauser, *Lectures on Numerical Mathematics*, Birkhäuser, Boston, 1990.
- [11] R. Zippel, Probabilistic algorithms for sparse polynomials, in: *Proc. EUROSAM'79*, in: *LNCS*, vol. 72, Springer, 1979.