

# On the Equivalence Between Prony's and Ben-Or's/Tiwari's Methods

MARK GIESBRECHT\*, GEORGE LABAHN† AND WEN-SHIN LEE‡

*Symbolic Computation Group  
School of Computer Science, University of Waterloo  
Waterloo, Ontario N2L 3G1, Canada*

June 2002

## Abstract

We show the equivalence between the exact Ben-Or/Tiwari algorithm and numerical Prony's method. Taking advantage of the recent results in both exact and numerical approaches, we present new algorithms and outline possible developments.

## 1. Introduction

In 1988 Ben-Or and Tiwari gave an exact multivariate interpolation algorithm that interpolates all the terms at once and is sensitive to the number of terms (Ben-Or and Tiwari, 1988). At the same time, the Prony's method (de Prony, 1795) has long been existed as a numerical algorithm fitting a sum of exponential functions.

We present the equivalence between the Prony's method and Ben-Or/Tiwari algorithm. Some results built upon Ben-Or/Tiwari algorithm have been extended to a sum of exponential functions. Finally, we outline our current work that is based on the recent progress on both algorithms.

## 2. Ben-Or/Tiwari algorithm and its early termination

Consider the polynomial

$$f(x_1, \dots, x_n) = \sum_{i=1}^t c_i x_1^{e_{1,i}} \cdots x_n^{e_{n,i}} \text{ with } c_i \neq 0.$$

\*URL: <http://www.uwaterloo.ca/~mwg> E-mail: [mwg@uwaterloo.ca](mailto:mwg@uwaterloo.ca)

†URL: <http://www.uwaterloo.ca/~glabahn> E-mail: [glabahn@uwaterloo.ca](mailto:glabahn@uwaterloo.ca)

‡URL: <http://www.wen-shin.com> E-mail: [ws2lee@scg.uwaterloo.ca](mailto:ws2lee@scg.uwaterloo.ca)

Let  $\beta_i(x_1, \dots, x_n) = x_1^{e_{1,i}} \cdots x_n^{e_{n,i}}$  be terms in  $f$ , and  $b_i$  denote  $\beta_i$  evaluated at distinct primes  $p_1, \dots, p_n$ , that is,  $b_i = \beta_i(p_1, \dots, p_n) = p_1^{e_{1,i}} \cdots p_n^{e_{n,i}}$ . Define an auxiliary polynomial  $\Lambda(\theta)$ :

$$\Lambda(\theta) = \prod_{i=1}^t (\theta - b_i) = \theta^t + \lambda_{t-1} \theta^{t-1} + \cdots + \lambda_0.$$

Ben-Or and Tiwari gave a multivariate interpolation algorithm that is sensitive to the number of terms, and thus efficient when the target polynomial is sparse.

**THEOREM 2.1 (BEN-OR/TIWARI ALGORITHM):** *For  $i \geq 0$ ,  $a_i = f(p_1^i, \dots, p_n^i)$  where  $p_1, \dots, p_n$  are distinct primes, the sequence  $\{a_i\}_{i \geq 0}$  is linearly generated by the auxiliary polynomial  $\Lambda(\theta)$ . Moreover,  $\Lambda(\theta)$  is the minimal linear generator of  $\{a_i\}_{i \geq 0}$  (Ben-Or and Tiwari, 1988).*

### The Ben-Or/Tiwari sparse interpolation algorithm

#### Input:

- $f$ :  $f = f(x_1, \dots, x_n)$ , a multivariate black box polynomial.
- $\sigma$ :  $\sigma \geq t$ ,  $t$  the number of non-zero terms.

#### Output:

$c_i$  and  $\beta_i$ :  $f = \sum_{i=1}^t c_i \beta_i$ .

1. (The Berlekamp/Massey algorithm.)  
 $a_i = f(p_1^i, \dots, p_n^i)$ ,  $0 \leq i \leq 2\sigma - 1$ ,  $p_1, \dots, p_n$  are distinct primes.  
 Compute  $\Lambda(\theta)$  from  $\{a_i\}_{0 \leq i \leq 2\sigma - 1}$ .
2. (Determine  $\beta_i$ .)  
 Find all  $t$  distinct roots of  $\Lambda(\theta)$ , which are  $b_i$ .  
 Determine each  $\beta_i$  through repeatedly divide  $b_i$  by  $p_1, \dots, p_n$ .
3. (Compute the coefficients  $c_i$ .)  
 solve a transposed Vandermonde system.

#### End.

The Ben-Or/Tiwari algorithm (Ben-Or and Tiwari, 1988) needs to know the number of terms,  $t$ , or an upper bound  $\sigma \geq t$ . Or we can guess  $\sigma$ , compute a candidate polynomial  $g$  for  $f$ , and compare  $g$  and  $f$  at an additional point. If  $f$  and  $g$  do not agree at the additional point, or we fail in computing  $g$ , then we can double the guess for  $\sigma$ .

Based on the strategy of “early termination” (Kaltofen et al., 2000), with high probability the interpolation can be finished within a single interpolation run.

**THEOREM 2.2 (EARLY TERMINATION BEN-OR/TIWARI ALGORITHM):**

*Pick  $p_1, \dots, p_n$  randomly and uniformly from a subset  $S$  of the domain of values, which is assumed to be an integral domain, then for the sequence  $\{a_i\}_{i \geq 1}$  with  $a_i = f(p_1^i, \dots, p_n^i)$ , the Berlekamp/Massey algorithm encounters  $\Delta = 0$  and  $i > 2L$  the first time at  $i = 2t + 1$  with probability no less than*

$$1 - \frac{t(t+1)(2t+1) \deg(f)}{6 \cdot \#(S)},$$

where  $\#(S)$  is the number of elements in  $S$  (Kaltofen et al., 2000).

In the implementation, the user can supply an integer  $\zeta \geq 1$ , and the early termination is triggered only after a zero discrepancy with  $i > 2L$  occurs  $\zeta$  times in a row. The higher thresholds can weed out bad random choices from sets that are much smaller than Theorem 2.2 would require. Nevertheless, the precise analysis for higher thresholds is complicated.

### The early termination Ben-Or/Tiwari algorithm

**Input:**

$f$ :  $f = f(x_1, \dots, x_n)$ , a multivariate black box polynomial.

$\zeta$ : a positive integer, the threshold for early termination.

**Output:**

$c_i$  and  $\beta_i$ :  $f = \sum_{i=1}^t c_i \beta_i$  with high probability.

Or an error message: if the procedure fails to complete.

1. (The early termination within the Berlekamp/Massey algorithm.)

*Pick random elements:  $p_1, \dots, p_n \notin \{0, 1\}$ .*

*For  $i = 1, 2, \dots$*

*$a_i = f(p_1^i, \dots, p_n^i)$ ;*

*Perform the Berlekamp/Massey algorithm on  $a_1, \dots, a_i$ .*

*If  $\Delta_i = 0$  and  $i > 2L$  happens  $\zeta$  times in a row, then*

*$\Lambda(\theta)$  is determined; break out of the loop;*

2. (Determine  $\beta_j$ .)

*Compute all the roots of  $\Lambda(\theta)$  in the domain of  $p_i$ .*

*If  $\Lambda(z)$  does not completely factor, or not all the roots are distinct, then the early termination was false.*

*Else, determine terms  $\beta_j$  from the roots of  $\Lambda(\theta)$ , which are  $b_j$ :*

*repeatedly divide  $b_j$  by  $p_1, \dots, p_n$ . Again, the term recovery might fail for unlucky  $p_1, \dots, p_n$ .*

3. (Determine  $c_j$ .)

*solve a transposed Vandermonde system.*

**End.**

### 3. The Prony's method and Ben-Or/Tiwari algorithm

In 1795, Gaspard Clair Francois Marie Riche de Prony gave a two-step method (de Prony, 1795) for fitting a sum of  $t$  exponential functions of the form:

$$f(x) = c_1 e^{\mu_1 x} + c_2 e^{\mu_2 x} + \cdots + c_t e^{\mu_t x}.$$

Knowing that  $f(x)$  is a sum of  $t$  exponential functions and having no less than  $2t$  evaluations of  $f(x)$  at equally spaced points, Prony's method determines the coefficients  $c_i$  and the bases in terms of  $\mu_i$ , for  $1 \leq i \leq t$ . We notice that Prony's method is closely related to the modern Ben-Or/Tiwari multivariate sparse interpolation algorithm (Ben-Or and Tiwari, 1988), in which the target polynomial is evaluated at powers of a point so that the terms in the polynomial would behave like exponential functions. For comparison, we outline Prony's method in parallel with the Ben-Or/Tiwari algorithm in Table 1.

**Table 1:** Comparison between Prony's method and Ben-Or/Tiwari algorithm

Prony's method	Ben-Or/Tiwari algorithm
Interpolate: $f(x) = \sum_{i=1}^t c_i e^{\mu_i x}$	Interpolate: $f(x_1, \dots, x_n) = \sum_{i=1}^t c_i x_1^{e_{1,i}} \cdots x_n^{e_{n,i}}$
Evaluate: $f(0), f(1), \dots, f(2t-1)$	Evaluate: $f(p_1^0, \dots, p_n^0), f(p_1^1, \dots, p_n^1), \dots, f(p_1^{2t-1}, \dots, p_n^{2t-1})$ , where $p_1, \dots, p_n$ are distinct primes.
1. Solve $\lambda_i$ in the $t \times t$ system: $\sum_{i=0}^{t-1} \lambda_i f(i+j) = -f(t+j)$ , and $j = 0, \dots, t-1$ . 2. Solve: $\Lambda(\theta) = \theta^t + \lambda_{t-1} \theta^{t-1} + \cdots + \lambda_0 = 0$ . The $t$ distinct roots are $e^{\mu_i}$ , $1 \leq i \leq t$ .	1. Compute <sup>†</sup> the minimal generating polynomial $\Lambda(\theta)$ of sequence $\{f(p_1^i, \dots, p_n^i)\}_{i=0}^{2t-1}$ . 2. Solve: $\Lambda(\theta) = \theta^t + \lambda_{t-1} \theta^{t-1} + \cdots + \lambda_0 = 0$ . The $t$ distinct roots are $p_1^{e_{1,i}} \cdots p_n^{e_{n,i}}$ , $1 \leq i \leq t$ .
3. Determine $c_j$ from $e^{\mu_j}$ and evaluations of $f$	3. Determine $c_j$ from $p_1^{d_{1,j}} \cdots p_n^{d_{n,j}}$ and evaluations of $f$

<sup>†</sup> Berlekamp/Massey algorithm

**Remark:** In Prony's method,  $f$  can be evaluated at any  $2t$  equally spaced points. Also, so far we have only seen  $\lambda_i$  in Prony's method being computed through either solving a linear system or least square methods (in the approximate case). Nevertheless, clearly in the exact case the Berlekamp/Massey algorithm can be utilized in locating  $\lambda_i$ .

### 4. Applications of sparse interpolation to Prony's method

Due to the similarities between Prony's method and the Ben-Or/Tiwari algorithm, some recent results in sparse polynomial interpolations are immediate to sums of exponential functions.

#### 4.1. The early termination of Prony's method

Consider a black box  $f(x)$  that is a sum of exponential functions:

$$f(x) = \sum_{i=1}^t c_i e^{\mu_i x}. \quad (1)$$

Without knowing  $t$ , with high probability we still can interpolate  $f(x)$ : randomly pick  $p$  and  $s \neq 0$ ; perform the Berlekamp/Massey algorithm on sequence  $f(p+s), f(p+2s), \dots, f(p+i \cdot s), \dots$ . Then with high probability when  $i = 2t+1$  the first zero discrepancy at  $i > 2L$  occurs. This is based on early termination in the Ben-Or/Tiwari algorithm (Theorem 2.2). Therefore, without knowing  $t$ , we can probabilistically interpolate the sum of exponential functions in (1) after  $2t + \zeta$  evaluations.

#### The early termination Prony's method

**Input:**

$f$ :  $f(x) = \sum_{i=1}^t c_i e^{\mu_i x}$ , a black box function.

$\zeta$ : a positive integer, the threshold for early termination.

**Output:**

$c_i$  and  $\mu_i$ :  $f = \sum_{i=1}^t c_i e^{\mu_i x}$  with high probability.

Or an error message: if the procedure fails to complete.

1. (The early termination within the Berlekamp/Massey algorithm.)

*Pick random elements:  $p, s \neq 0$ .*

*For  $i = 1, 2, \dots$*

*$a_i = f(p + i \cdot s)$ ;*

*Perform the Berlekamp/Massey algorithm on  $a_1, \dots, a_i$ .*

*If  $\Delta_i = 0$  and  $i > 2L$  happens  $\zeta$  times in a row, then*

*$\Lambda(\theta)$  is determined; break out of the loop;*

2. (Determine  $\mu_j$ .)

*Compute all the roots of  $\Lambda(\theta)$ .*

*If  $\Lambda(\theta)$  does not completely factor, or not all the roots are distinct, then the early termination was false.*

*Else, determine  $\mu_j$  from the roots of  $\Lambda(\theta)$ , which are  $e^{\mu_j}$ .*

3. (Determine  $c_j$ .)

*solve a transposed Vandermonde system.*

**End.**

## 4.2. The multivariate Prony's method

We now consider a multivariate black box  $f(x_1, \dots, x_n)$  that is a sum of exponential functions:

$$f(x_1, \dots, x_n) = \sum_{i=1}^t c_i e^{\mu_{1,i}x_1 + \mu_{2,i}x_2 + \dots + \mu_{n,i}x_n}. \quad (2)$$

Here we implement a variable by variable approach similar to Zippel's method (Zippel, 1979) and the prunings via early termination (Kaltofen et al., 2000). By the early termination Prony's method, we first interpolate  $f(x_1, p_2, \dots, p_n)$  in  $x_1$  with  $(x_2, \dots, x_n)$  fixed at random  $(p_2, \dots, p_n)$ . If  $f$  is correctly interpolated in  $x_1$ , we obtain  $f_1(x_1) = \sum_i c_{1,i} e^{\mu_{1,i}x_1}$ , where each  $c_{1,i}$  is

$$\sum_{a_{1,r}=a_{1,i}} c_r e^{a_{2,r}x_2 + \dots + a_{n,r}x_n} \text{ evaluated at } (x_2, \dots, x_n) = (p_2, \dots, p_n).$$

We then continue the interpolation of each  $c_{1,i}$  in  $x_2$  and obtain  $f_2(x_1, x_2)$ . Assume we obtain the correct result at every stage, which is with high probability, finally  $f_n$  interpolate (2). For reference, we outline the algorithm steps.

### The multivariate Prony algorithm with early termination

#### Input:

$f$ : a black box function that is a sum of exponential functions.

$(x_1, \dots, x_n)$ : an ordered list of variables in  $f$ .

$\zeta$ : a positive integer, the threshold for early termination.

#### Output:

$\sum_{i=1}^t c_i e^{\mu_{1,i}x_1 + \mu_{2,i}x_2 + \dots + \mu_{n,i}x_n}$ : which equals  $f$  with high probability.

Or an error message: if the procedure fails.

1. (Initialize the anchor points.)  
Randomly pick  $p_1, \dots, p_n$  and  $s_1 \neq 0, \dots, s_n \neq 0$ ;
2. (Interpolate one more variable: with high probability, we have  $f_{j-1} = f(x_1, \dots, x_{j-1}, p_j, \dots, p_n) = \sum_{r \in K_{j-1}} c_{j-1,r} e^{a_{1,r}x_1 + \dots + a_{j-1,r}x_{j-1}}$ .)  
For  $j = 1, \dots, n$  Do  
     $k_{j-1} \leftarrow \#(K_{j-1})$
3. (Interpolate every  $c_{j-1,r}$  in  $x_j$  by early termination Prony's method. Locate the value of every such coefficient at  $x_j = p_j + i \cdot s_j$  by solving a  $k_{j-1}$  by  $k_{j-1}$  transposed Vandermonde system.)  
For  $i = 1, \dots, \text{while not all } c_{j-1,r}(x_j) \text{ are interpolated do}$   
    For  $r = 1, \dots, k_{j-1}$  Do

$$\begin{aligned} & \sum_{r \in K_{j-1}} c_{j-1,r}^{[i]} e^{\mu_{1,r}(\tilde{p}_1 + r\tilde{s}_1) + \dots + \mu_{j-1,r}(\tilde{p}_{j-1} + r\tilde{s}_{j-1})} \\ & = f(\tilde{p}_1 + r\tilde{s}_1, \dots, \tilde{p}_{j-1} + r\tilde{s}_{j-1}, p_j + i s_j, p_{j+1}, \dots, p_n); \end{aligned}$$

*If the system is singular then report "Failure;"*  
*Else solve for all  $c_{j-1,r}^{[i]}$ ;*

4. (Interpolate  $k_{j-1}$  many sums of exponential functions in  $x_j$ . They are the coefficients for exponential functions in  $x_1, \dots, x_{j-1}$ .)

*For every  $r \in K_{j-1}$  Do*

*the early termination Prony's method on  $\{c_{j-1,r}^{[1]}, \dots, c_{j-1,r}^{[i]}\}$ ;*  
*if early termination occurs:  $c_{j-1,r}(x_j) \leftarrow \sum_{\tilde{r}} c_{j-1,r,\tilde{r}} e^{\mu_{j,\tilde{r}} x_j}$ ;*

5. (Update  $K_j$ .)

$K_j = \emptyset$ ;  $r^{new} = 1$ ;

*For every  $r \in K_{j-1}$  and  $\tilde{r}$  Do*

*If  $c_{j-1,r,\tilde{r}} \neq 0$  then*

$c_{j,r^{new}} \leftarrow c_{j-1,r,\tilde{r}}$ ;  $K_j \leftarrow K_j \cup \{r^{new}\}$ ;  $r^{new} \leftarrow r^{new} + 1$ ;

*Randomly pick  $\tilde{p}_j, \tilde{s}_j \neq 0$ ;*

**End.**

### 4.3. Applications to a linear ordinary differential equations

Now we again consider a univariate black box  $f(x)$  that is a sum of exponential functions and its derivatives:

$$\begin{aligned} f(x) &= \sum_{i=1}^t c_i e^{\mu_i x} \\ f^{(1)}(x) &= \sum_{i=1}^t \mu_i c_i e^{\mu_i x} \\ &\vdots \\ f^{(j)}(x) &= \sum_{i=1}^t \mu_i^j c_i e^{\mu_i x} \end{aligned}$$

Based on the early termination Ben-Or/Tiwari algorithm, for a random  $p$ , perform the Berlekamp/Massey algorithm on  $f^{(1)}(p), f^{(2)}(p), \dots, f^{(j)}(p), \dots$ , with high probability at  $j = 2t + 1$  we shall obtain  $\Lambda(\theta)$  whose  $t$  distinct roots are  $\mu_i$ . The recovery of  $c_i$  is based on locating  $c_i e^{\mu_i p}$  as the coefficients of  $\mu_i$  and that  $p$  and all  $\mu_i$  are known at this point. This provides an algorithm to interpolate a sum of exponential functions from evaluating its consecutive derivatives at a point.

Note that in this case, the auxiliary polynomial  $\Lambda(\theta)$  gives a linear ordinary differential equation such that  $f$  and its derivatives,  $f^{(i)}$  for  $i \geq 0$ , are all its solutions, that is, for  $i \geq 0$ :

$$\Lambda\left(\frac{d}{dx}\right)(f^{(i)}) = \frac{d^t}{dx^t} f^{(i)} + \lambda_{t-1} \frac{d^{t-1}}{dx^{t-1}} f^{(i)} + \dots + \lambda_1 \frac{d}{dx} f^{(i)} + \lambda_0 f^{(i)} = 0. \quad (3)$$

#### 4.4. Linear partial differential system

When  $f$  is given as

$$f(x_1, \dots, x_n) = \sum_{i=1}^t c_i e^{\mu_{1,i}x_1 + \mu_{2,i}x_2 + \dots + \mu_{n,i}x_n},$$

and

$$\Lambda_j(\theta) = \prod_{r=1}^{t_j} (\theta - \mu_{j,r}) = \theta^{t_j} + \lambda_{j,t_j-1} \theta^{t_j-1} + \dots + \lambda_{j,1} \theta + \lambda_{j,0},$$

then for every  $j$ ,  $\frac{d^i}{dx_j^i} f$  for all  $i \geq 0$  satisfy the equation in (4):

$$\Lambda_j\left(\frac{d}{dx_j}\right)\left(\frac{d^i f}{dx_j^i}\right) = \frac{d^t}{dx_j^t} \frac{d^i f}{dx_j^i} + \lambda_{j,t-1} \frac{d^{t-1}}{dx_j^{t-1}} \frac{d^i f}{dx_j^i} + \dots + \lambda_{j,0} \frac{d^i f}{dx_j^i} = 0. \quad (4)$$

**Remark:** Similar results to (4) and (3) can be extended to other linear differential operators, such as  $D_{x_1, x_2} = \frac{d^2}{dx_1 dx_2}$ .

## 5. Current developments

Although Prony's method has long been appeared as a numerical algorithm, it is unstable due to its root finding procedure. We note that certain least square approaches has been exploited in (Milanfar et al., 1995). On the other hand, by reformulating it as a generalized eigenvalue problem, Golub et al. (1999) gave a stable algorithm. Based on the recent numerical results in Prony's method, we intend to develop numerical sparse interpolation algorithms.

## References

- M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proc. Twentieth Annual ACM Symp. Theory Comput.*, pages 301–309, New York, N.Y., 1988. ACM Press.
- R. de Prony. Essai expérimental et analytique sur les lois de la dilabilité et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. *J. de l'École Polytechnique*, 1:24–76, 1795.
- G. H. Golub, P. Milanfar, and J. Varah. A stable numerical method for inverting shape from moments. *SIAM J. Sci. Comput.*, 21(4):1222–1243, 1999.
- E. Kaltofen, W.-s. Lee, and A. A. Lobo. Early termination in Ben-Or/Tiwari sparse interpolation and a hybrid of Zippel's algorithm. In C. Traverso, editor, *Internat. Symp. Symbolic Algebraic Comput. ISSAC 2000 Proc. 2000 Internat. Symp. Symbolic Algebraic Comput.*, pages 192–201, New York, N. Y., 2000. ACM Press.



- P. Milanfar, G. C. Verghese, W. C. Karl, and A. S. Willsky. Reconstructing polygons from moments with connections to array processing. *IEEE Transactions on Signal Processing*, 43(2):432–443, 1995.
- R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. EUROSAM '79*, volume 72 of *Lect. Notes Comput. Sci.*, pages 216–226, Heidelberg, Germany, 1979. Springer Verlag.