Early Termination Strategies in Sparse Interpolation Algorithm

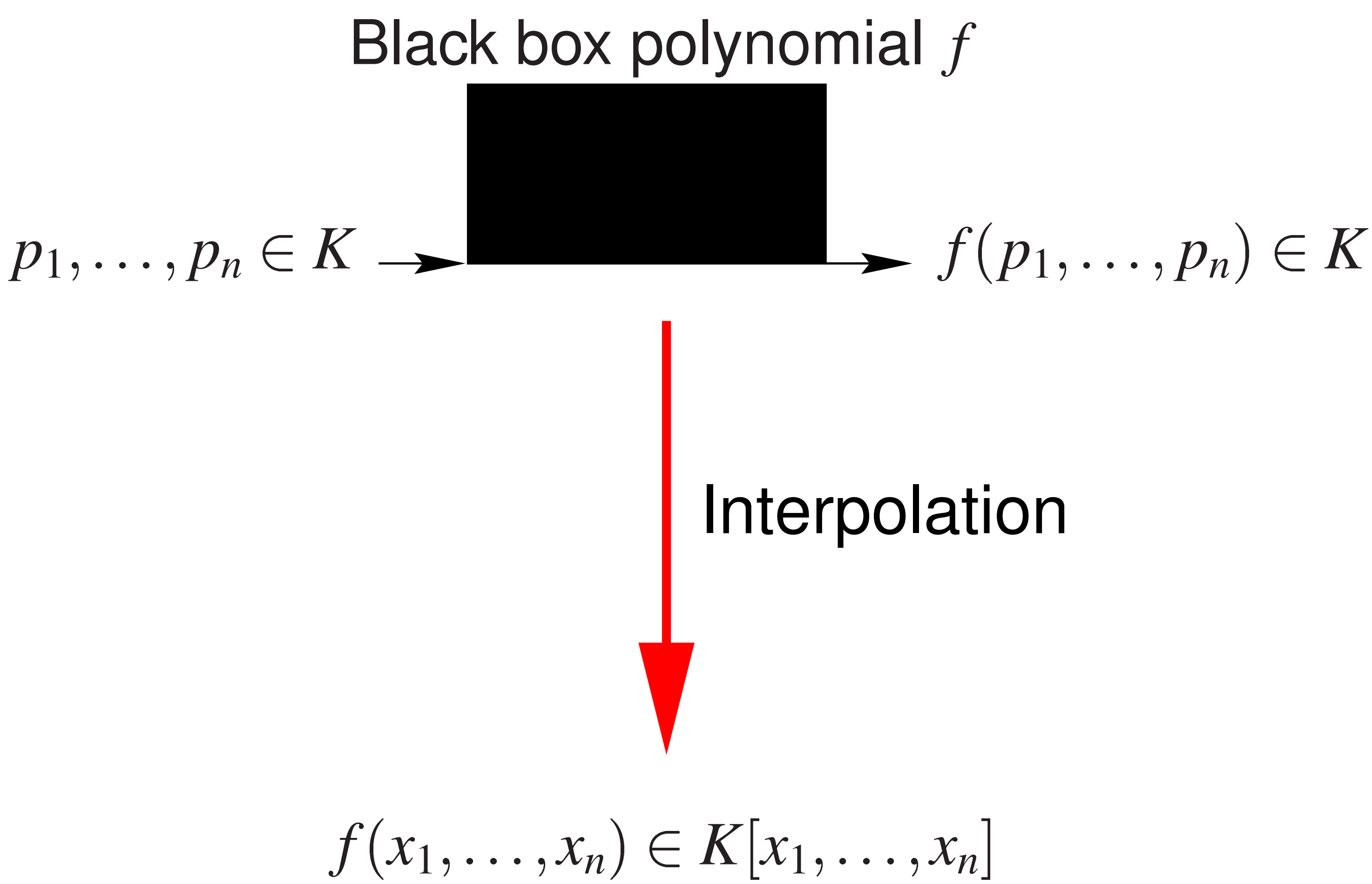
Wen-shin Lee
North Carolina State University
www.wen-shin.com

June 7, 2000

Outline

Objective	Black box polynomial interpolation. Early termination in sparse interpolation. Why early termination?
Previous Research	Zippel's probabilistic algorithm. Berlekamp/Massey algorithm. Ben-Or/Tiwari's deterministic algorithm.
Current Research	Early termination. Idea of hybrid. Pruning. Maple implementation. Thresholds. Moduli.
Future Research	Implement sparse shifts in Maple. Early termination on sparse shifts. Probability analysis for thresholds. Very small coefficient fields.

Objective Black box polynomial interpolation.



Objective Early termination in sparse interpolation.

The idea of early termination is a general paradigm that has been used in the setting of Wiedemann algorithm.

When the black box polynomial is sparse, $f(x_1,\ldots,x_n)$ can be probabilistically determined from its values at fewer points. That is, provided the correctness is probabilistic, the computational complexities are sensitive to the sparsity of the target polynomial.

Objective Why early termination?

- Save time and space.

- A useful tool for controlling intermediate expression swell in computer algebra.

- Adapt to the sparsity of the target polynomial as the interpolation proceeds, even when not much information is provided for the black box polynomial, such as bounds on degree or number of terms.

Previous Research Zippel's probabilistic algorithm (1979).

During the variable by variable interpolation, a zero coefficient is the image of a zero polynomial with high probability.

Let $f \in K[x_1, \ldots, x_n]$ be non-zero, $S \subseteq K$:

$$\operatorname{Prob}(f(a_1, \ldots, a_n) \neq 0 \mid a_i \in S \subseteq K) \geq 1 - \frac{\deg(f)}{\operatorname{Cardinality}(S)}$$

Previous Research An example on Zippel's probabilistic algorithm.

Interpolate $f(x,y) = 3x^5y^3 + 2x^5 + y^2 + 5 \in K[x,y]$.

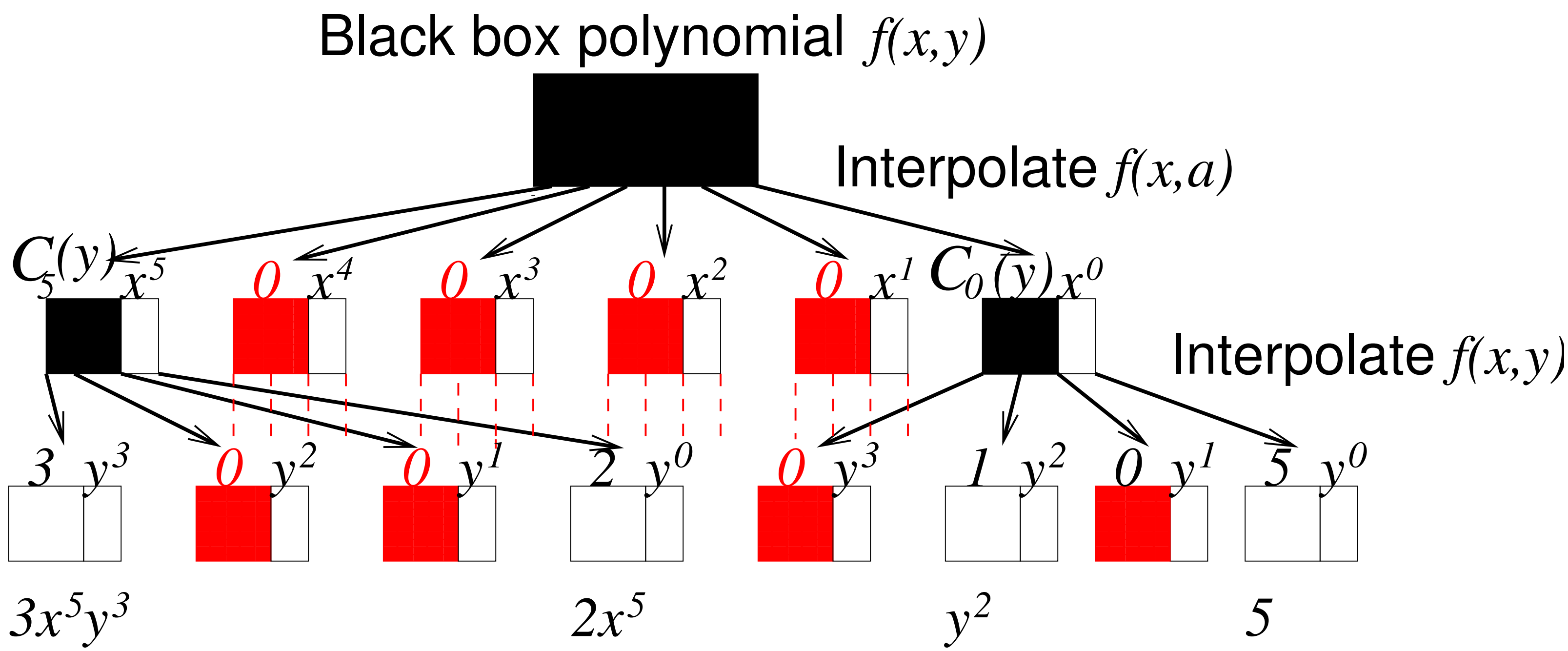
1. *Pick a random* $a \in S$*, interpolate* $f(x,a)$*.*

2. (Assume that x^4, x^3, x^2, x have a zero coefficient in y.)
   Interpolate $f(x,y) = C_5(y)x^5 + C_0(y)$*:*
   Pick $p, b_0, b_1, \ldots$ *and compute* $C_i(b_j)$ *from a transposed Vandermonde system.*

$$C_5(b_j)(p^5)^k + C_0(b_j) = f(p^k, b_j), k = 0, 1.$$

3. *Interpolate each* C_i*,* C_5 *and* C_0*.*

Previous Research An example on Zippel's probabilistic algorithm (figure).



Berlekamp/Massey algorithm.

Ben-Or and Tiwari (1988) proposed a deterministic sparse interpolation algorithm based on the Berlekamp/Massey algorithm.

(See examples in Maple worksheet.)

Previous Research Ben-Or/Tiwari's deterministic algorithm.

Input: f: a multivariate black box polynomial.
τ: $\tau \geq t$, t is the number of terms in f.
Output: c_j and m_j: $f = \sum_{j=1}^{t} c_j m_j$.

1. (The Berlekamp/Massey algorithm.)
   $a_i = f(p_1^i, \ldots, p_n^i)$, $0 \leq i \leq \tau$, p_i is the i-th prime.
   Compute $\Lambda(z)$ from $\{a_i\}_{\tau \geq i \geq 0}$.

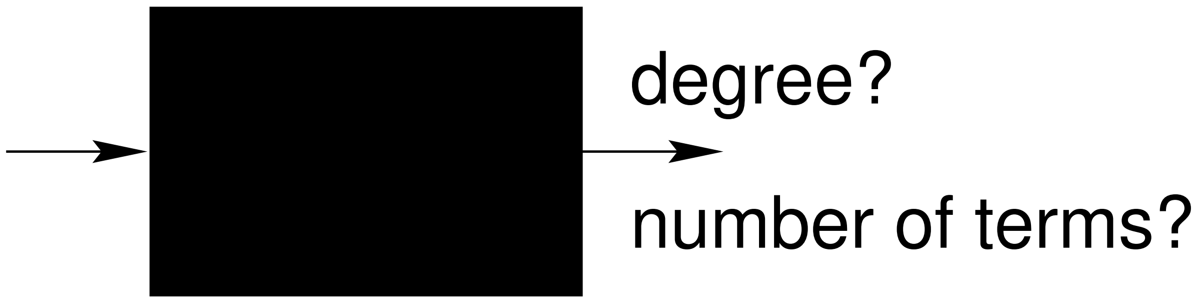
2. (Determine m_j.)
   Find all t distinct roots of $\Lambda(z)$, b_j.
   Determine all m_j: repeatly divide every b_j by $p_1, \ldots, p_n$.

3. (Compute c_j.)
   Solve a transposed Vandermonde system.

Previous Research: Recovery of term exponents in Ben-Or/Tiwari.

Multivariate terms [Kaltofen, Lakshman, Wiley 1990]	Univariate terms
1. *Use prime numbers* $p_1, \ldots, p_n$.	1. *Can use the smallest prime,* 2.
2. *Factor* $\Lambda(z) \bmod q$, *where* $q > \max_{\mathbf{e}} p_1^{e_1} \cdots p_n^{e_n}$.	2. *Factor* $\Lambda(z) \bmod q$, *where* $q > 2^{\deg(f)}$.
3. *Decompose term values.*	3. *Decompose term values.*

Early termination.



What if we do not know the upper bound of degree or number of terms of the target black box polynomial?

- Guess (and double the guess if fails.)
- Early termination.

Current Research Early termination in Newton interpolation.

For $i \leftarrow 1, 2, \ldots$ *Do*

Pick random p_i *and from* $f(p_i)$

compute

$$f^{[i]}(x) \leftarrow c_0 + c_1(x - p_1) + c_2(x - p_1)(x - p_2) + \cdots$$
$$\equiv f(x) \pmod{(x - p_1)\cdots(x - p_i)}$$

If $c_i = 0$ *stop.*

End For

Threshold η:
In order to obtain a better probability, we require $c_i = 0$ more than once before terminating.

Current Research The early termination of Ben-Or/Tiwari's algorithm.

If $p_1, \ldots, p_n$ are chosen randomly and uniformly from a subset S of the domain of values then for the sequence

$$a_i = f(p_1^i, \ldots, p_n^i), i = 1, 2, \ldots$$

the Berlekamp/Massey algorithm encounters $\Delta = 0$ and $2L < r$ the first time for $r = 2t + 1$ with probability no less than

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

where t is the number of terms of f.

Threshold ζ:

In order to obtain a better probability, we require $\Delta = 0$ when $2L < r$ more than once before terminating.

Current Research The early termination interpolation in non-standard bases.

Early termination theorem has been proved for the following nonstandard bases:

- Pochhammer basis:

$$f(x) = \sum_{k=1}^{t} f_k x^{\bar{e}_k},$$

where $f_k \in K$ and $x^{\bar{n}} = x(x+1)\ldots(x+n-1)$.

- Chebyshev basis:

$$f(x) = \sum_{k=1}^{t} f_k T_{e_k}(x),$$

where $f_k \in K$, $T_0(x) = 1$, $T_1(x) = x$, and
$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$ for $n \geq 2$.

Current Research Compare early termination in Newton and Ben-Or/Tiwari.

Zippel's algorithm weeds out zero coefficient terms and proceeds one variable at a time, but each new variable is still interpolated densely.

Compare different early termination interpolation algorithms on $f(x_1,\ldots,x_n)$ with t its number of terms:

Algorithm	Ben-Or/Tiwari (Multivariate)	Ben-Or/Tiwari (Univariate)	Newton (Univariate)
Modulus (assume coefficients captured)	for recovery of all terms ($q > \max_{\mathbf{e}} p_1^{e_1} \cdots p_n^{e_n}$)	for recovery of all terms w.r.t. any variable ($q > 2^{\deg(f)}$)	for randomization ($q > \deg(f) + \eta$)
Black box probes	$2t + \zeta$	$\mathrm{O}(tn)$	$\mathrm{O}(n \deg(f))$
Computation result	might not finish	might not finish	always finishes

Current Research

Idea of hybrid.

Embed the univariate Ben-Or/Tiwari in Zippel; guard against failure by concurrent Newton on the same black box probes. We race Newton against Ben-Or/Tiwari in univariate interpolations within Zippel's algorithm.

Current Research — Why hybrid?

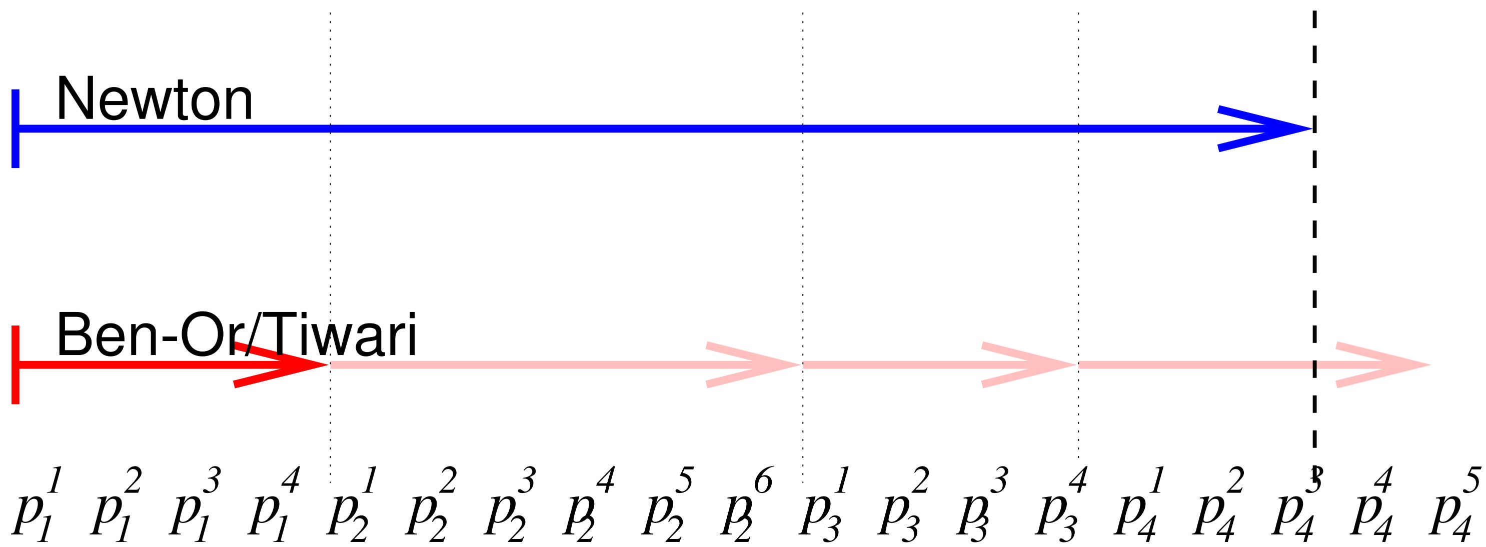
We can take advantage of both algorithms, or compensate for the disadvantage of either one:

- Terminate earlier when there are few terms.
  (Ben-Or/Tiwari $+$)

- Newton might outrace Ben-Or/Tiwari. (Newton $+$)

- Some information acquired from one algorithm can be used to check the result of the other algorithm, e.g., to see whether $\deg(f)$ recovered from Ben-Or/Tiwari is less than or equal to the degree of the most Newton interpolant.

Current Research A likely racing scenario in hybrid.



Delay the update of Newton interpolant:

When $p_i^r = p_j^s$, Newton might falsely early terminate. We delay the update at a repeated point and hence improve the probability of success.

Current Research — Term pruning during Zippel.

In Zippel's variable by variable interpolation, the monomials with zero coefficients are pruned. To further reduce the intermediate systems and expressions, we do two other kinds of pruning.

- Temporary pruning.
- Permanent pruning.

Current Research Temporary pruning.

During the variable by variable interpolation, we interpolate the coefficient polynomials $C_j^i(x_i)$ in $K[x_i]$. Polynomials C_j^i are coefficients of terms in $x_1, \ldots, x_{i-1}$:

$$f(x_1, \ldots, x_{i-1}, x_i, a_{i+1}, \ldots, a_n) = \sum_j C_j^i(x_i) x_1^{e_{j,1}} \cdots x_{i-1}^{e_{j,i-1}}.$$

Some of those coefficient polynomials might be determined early via early termination, but the corresponding terms are not fully determined. We may prune those terms *temporarily* and reduce the dimension of the transposed Vandermonde system before all the other C_j^i are determined.

Permanent pruning.

Introduce the *homogenizing variable* x_0 and define $\tilde{f}(x_0, x_1, \ldots, x_n)$ as follows:

$$\tilde{f}(x_0, x_1, \ldots, x_n) = f(x_0 x_1, \ldots, x_0 x_n) = \sum_i c_i x_0^{e_{i,1}+\cdots+e_{i,n}} x_1^{e_{i,1}} \cdots x_n^{e_{i,n}}.$$

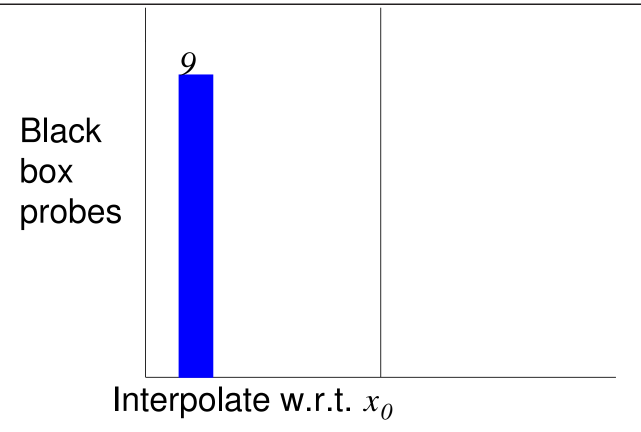
Interpolate $\tilde{f}$ with respect to x_0 at random $a_1, \ldots, a_n$ as $f_0(x_0) = \tilde{f}(x_0, a_1, \ldots, a_n)$. By comparing the total degree of each term, the degree of x_0, we can tell when a term is complete during interpolation. Such a term can then be *permanently pruned* from the all the remaining interpolation process and reduce all the following transposed Vandermonde systems.

[Díaz and Kaltofen 1998]

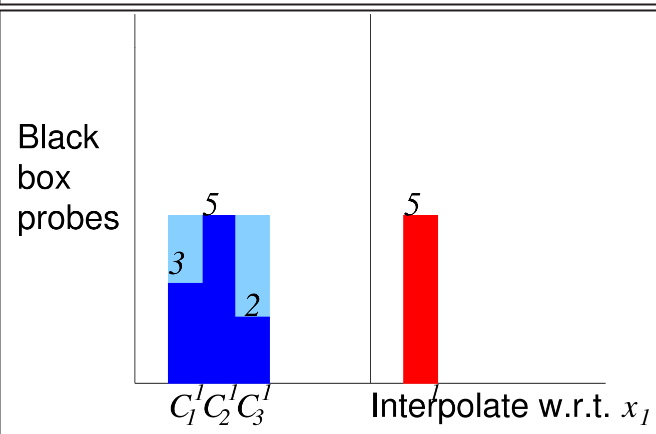
Current Research Compare permanent pruning and temporary pruning in an example (hybrid algorithm implemented).

Interpolate $f(x_1,x_2,x_3,x_4,x_5) = 3x_1^5x_2^3 + 2x_1^5 + x_2^2 + x_4^5 + 5$:

Introduce x_0 and interpolate $\tilde{f}$ w.r.t. x_0:
$\tilde{f}(x_0,a_1,\ldots,a_5)$
$= f(x_0a_1,\ldots,x_0a_5)$



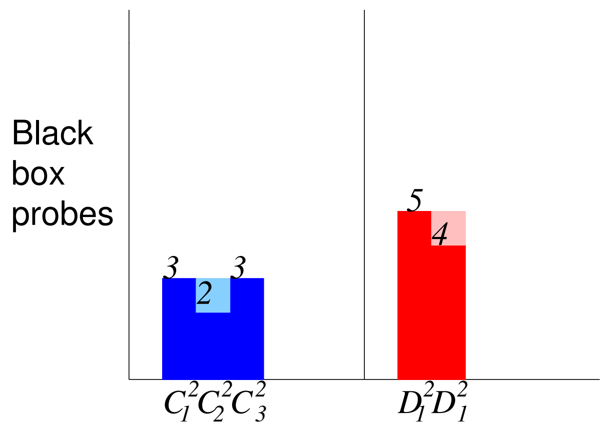
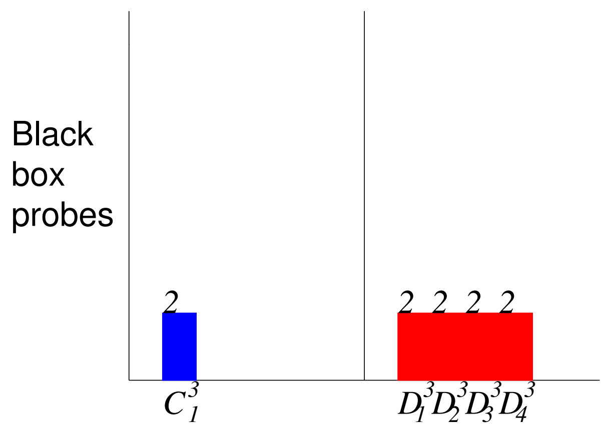
Interpolate C_1^1, C_2^1, C_3^1 w.r.t. x_1:
$\tilde{f}(x_0,x_1,a_2,\ldots,a_5)-5$
$=C_1^1(x_1)x_0^8+C_2^1(x_1)x_0^5$
$+C_3^1(x_1)x_0^2$



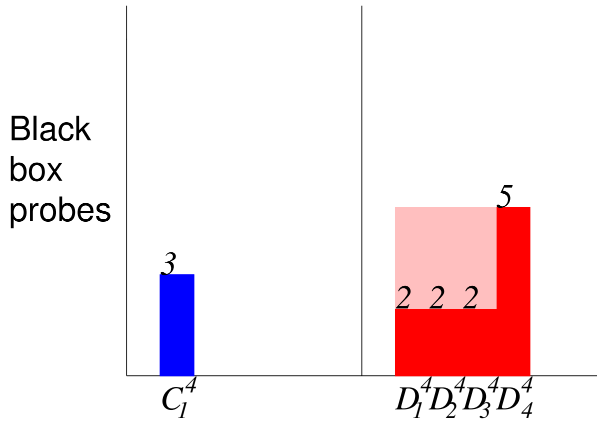
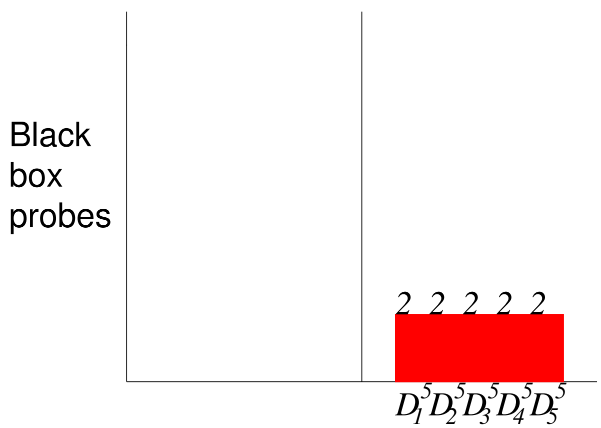
Interpolate $f(x_1,a_2,\ldots,a_5)$ w.r.t. x_1:

Current Research

Compare permanent pruning and temporary pruning in an example (hybrid algorithm implemented).

Interpolate C_1^2, C_2^2, C_3^2 w.r.t. x_2: $\tilde{f}(x_0,x_1,x_2,a_3,a_4,a_5)$ $-5-2x_1^5$ $= C_1^2(x_2)x_0^8x_0^5$ $+C_2^2(x_2)x_0^5+C_3^2(x_2)x_0^2$		Interpolate D_1^2, D_2^2 w.r.t. x_2: $f(x_1,x_2,a_3,\ldots,a_5)$ $= D_1^2(x_2)x_1^5+D_2^2(x_2)$
Interpolate C_1^3 w.r.t. x_3: $\tilde{f}(x_0,\ldots,x_3,a_4,a_5)$ $-5-2x_1^5-3x_1^5x_2^3$ $-x_2^2= C_1^3(x_3)x_0^5$		Interpolate D_1^3, D_2^3, D_3^3, D_4^3 w.r.t. x_3: $f(x_1,\ldots,x_3,a_4,a_5)$ $= D_1^3(x_3)x_1^5x_2^3$ $+D_2^3(x_3)x_1^5+D_3^3(x_3)x_2^2$ $+D_4^3(x_3)$

Compare permanent pruning and temporary pruning in an example (hybrid algorithm implemented).

Interpolate C_1^4 w.r.t. x_4: $\tilde{f}(x_0,\ldots,x_4,a_5)$ $-5-2x_1^5-3x_1^5x_2^3-x_2^2$ $=C_1^4(x_4)x_0^5$		Interpolate D_1^4, D_2^4 D_3^4, D_4^4 w.r.t. x_4: $f(x_1,\ldots,x_4,a_5)$ $=D_1^4(x_4)x_1^5x_2^3$ $+D_2^4(x_4)x_1^5+D_3^4(x_4)x_2^4$ $+D_4^4(x_4)$
All the terms have been permanently pruned.		Interpolate $D_1^5, D_2^5,$ D_3^5, D_4^5, D_5^5 w.r.t. x_5: $f(x_1,\ldots,x_5)$ $=D_1^5(x_5)x_1^5x_2^3$ $+D_2^5(x_5)x_1^5+D_3^5(x_5)x_2^2$ $+D_4^5(x_5)x_4^5+D_5^5(x_5)$
32 black box probes		43 black box probes

Maple implementation.

The *protobox* package is our Maple V.5.1 implementation of this new hybrid algorithm.

Current Research Compare different embedded univariate interpolations and the ”racing” algorithm in *protobox*.

Either Newton or Ben-Or/Tiwari can be ”turned off” by setting its threshold to ∞ and thus all the interpolations will be forced to be performed in the remaining active one.

The performance of the hybrid algorithm, or ”racing” algorithm, always takes advantage of both Newton and Ben-Or/Tiwari; the average black box probes needed is never more than the minimum of either one.

Current Research Compare different embedded univariate interpolations and the "racing" algorithm in *protobox*.

$$
\begin{aligned}
f_1(x_1,\ldots,x_{10}) = {} & x_1^2x_3^3x_4x_6x_8x_9^2 + x_1x_2x_3x_4^2x_5^2x_8x_9 + x_2x_3x_4x_5^2x_8x_9 + x_1x_3^3x_4^2x_5^2x_6^2x_7x_8^2 + x_2x_3x_4x_5^2x_6x_7x_8^2 \\
f_2(x_1,\ldots,x_{10}) = {} & x_1x_2^2x_4^2x_8x_9^2x_{10}^2 + x_2^2x_4x_5^2x_6x_7x_9x_{10}^2 + x_1^2x_2x_3x_5^2x_7^2x_9^2 + x_1x_3^2x_4^2x_7^2x_9^2 + x_1^2x_3x_4x_7^2x_8^2 \\
f_3(x_1,\ldots,x_{10}) = {} & 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 \\
& +12x_2^3x_4^3x_6^3x_7^2x_8^3 \\
f_4(x_1,\ldots,x_{10}) = {} & 9x_1^2x_3x_4x_6^3x_7^2x_8x_{10}^4 + 17x_1^3x_2x_5^2x_6^2x_7x_8^3x_9^4x_{10}^3 + 17x_2^2x_3^4x_4^2x_7^4x_8^3x_9x_{10}^3 + 3x_1^3x_2^2x_6^3x_{10}^2 \\
& +10x_1x_3x_5^2x_6^2x_7^4x_8^4 \\
f_5(x_1,\ldots,x_{50}) = {} & \textstyle\sum_{i=1}^{50} x_i^{50} \\
f_6(x_1,\ldots,x_5) = {} & \textstyle\sum_{i=1}^{5} (x_1+x_2+x_3+x_4+x_5)^i \\
f_7(x_1,x_2,x_3) = {} & x_1^{20} + 2x_2 + 2x_2^2 + 2x_2^3 + 2x_2^4 + 3x_3^{20}
\end{aligned}
$$

	mod	Newton	Ben-Or/Tiwari	"Racing"
f_1	100003	147	137	126
f_2	100003	146	143	124
f_3	100003	209	143	133
f_4	100003	188	149	133
f_5	100000007†	2652	251	251
f_6	100000007†	965	1256	881
f_7	100003	94	46	41

Average number of black box probes needed for different embedded univariate interpolation algorithms after 10 runs. Newton and Ben-Or/Tiwari thresholds are both default as 1.

† This is tested in Maple 6

Thresholds implemented in *protobox*.

Improve the probability of correctness:

η: (default 1) Newton interpolation threshold.

ζ: (default 1) Ben-Or/Tiwari interpolation threshold.

τ: (default 0) number of points used for post test. Check whether the interpolating polynomial and the input black box agree at a few random points before the result to be returned.

Thresholds implemented in *protobox*.

Improve the throughput of overall algorithm:

κ: (default 0) number of random numbers retried before abort the interpolation if two terms map to a same value and cause a singular Vandermonde system.

γ: (default 0) extends the upper bound of each univariate interpolation loop. This regards the delay in updating Newton interpolants.

Current Research Throughputs under different modulus and thresholds.

$$
\begin{aligned}
f_1 &= x_1^2x_3^3x_4x_6x_8x_9^2 + x_1x_2x_3x_4^2x_5^2x_8x_9 + x_2x_3x_4x_5^2x_8x_9 + x_1x_3^3x_4^2x_5^2x_6^2x_7x_8^2 + x_2x_3x_4x_5^2x_6x_7x_8^2 \\
f_2 &= x_1x_2^2x_4^2x_8x_9^2x_{10}^2 + x_2^2x_4x_5^2x_6x_7x_9x_{10}^2 + x_1^2x_2x_3x_5^2x_7^2x_9^2 + x_1x_3^2x_4^2x_7^2x_9^2 + x_1^2x_3x_4x_7^2x_8^2 \\
f_3 &= 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3 \\
f_4 &= 9x_1^2x_3x_4x_6^3x_7^2x_8x_{10}^4 + 17x_1^3x_2x_5^2x_6^2x_7x_8^3x_9^4x_{10}^3 + 17x_2^2x_3^4x_4^2x_7^4x_8^3x_9x_{10}^3 + 3x_1^3x_2^2x_6^3x_{10}^2 + 10x_1x_3x_5^2x_6^2x_7^4x_8^4
\end{aligned}
$$

	Thresholds			mod 31			mod 37			mod 41			mod 43			mod 47			mod 53		
	η, ζ	τ	κ, γ	$=$	$\neq$	!	$=$	$\neq$	!	$=$	$\neq$	!	$=$	$\neq$	!	$=$	$\neq$	!	$=$	$\neq$	!
f_1	1	0	0	8	2	90	7	1	92	15	3	82	11	5	84	25	3	72	20	2	78
	2	1	2	30	0	70	38	1	61	44	0	56	55	0	45	71	0	29	52	0	48
	3	2	4	38	0	62	36	0	64	50	0	50	60	0	40	79	0	21	70	0	30
f_2	1	0	0	4	3	93	4	3	93	5	3	92	7	5	88	22	4	74	23	1	76
	2	1	2	22	0	78	36	0	64	38	0	62	48	1	51	61	0	39	66	0	34
	3	2	4	41	0	59	45	0	55	51	0	49	57	0	43	83	0	17	81	0	19
f_3	1	0	0	0	2	98	0	6	94	3	3	94	4	0	96	6	5	89	9	1	90
	2	1	2	3	1	96	8	0	92	16	0	84	10	0	90	37	0	63	27	0	73
	3	2	4	9	0	91	8	0	92	26	0	74	15	0	85	52	0	48	54	0	46
f_4	1	0	0	1	4	95	0	2	98	4	2	94	8	3	89	18	2	80	5	3	92
	2	1	2	8	0	92	5	0	95	20	0	80	22	0	78	63	0	37	44	0	56
	3	2	4	10	0	90	10	0	90	33	0	67	32	0	68	80	0	20	47	0	53

Current Research

Interpolation on small moduli.

Consider polynomial $f_1 = x_1^2 x_3^3 x_4 x_6 x_8 x_9^2 + x_1 x_2 x_3 x_4^2 x_5^2 x_8 x_9 + x_2 x_3 x_4 x_5^2 x_8 x_9 + x_1 x_3^3 x_4^2 x_5^2 x_6^2 x_7 x_8^2 + x_2 x_3 x_4 x_5^2 x_6 x_7 x_8^2$, its degrees in every single variable are smaller then its total degree. That is,

$$\deg(f_1) = 13 > 3 = \max_i \deg(f_1(x_i)).$$

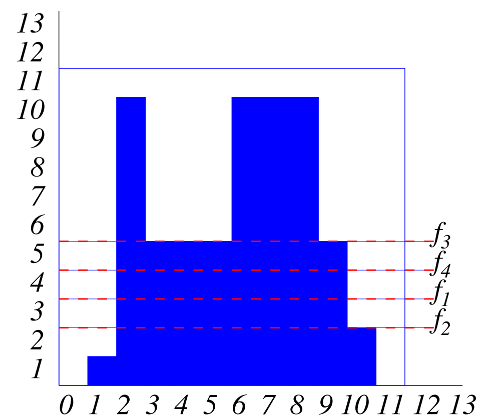
By ”turning off” homogenizing variable modification, f_1 can be interpolated on some small moduli q with $q \leq 13$ provided.

Current Research

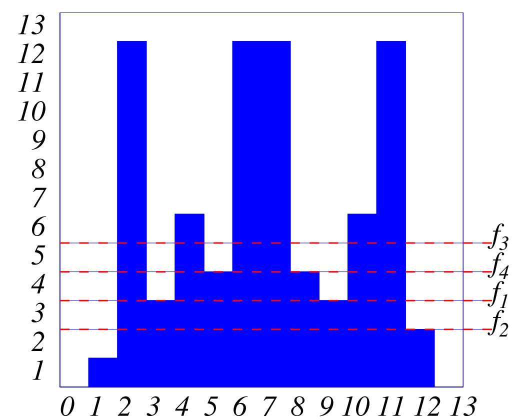
Performance on small moduli.

$$
\begin{aligned}
\deg(f_1) = 13: \quad f_1 &= x_1^2x_3^3x_4x_6x_8x_9^2 + x_1x_2x_3x_4^2x_5^2x_8x_9 + x_2x_3x_4x_5^2x_8x_9 + x_1x_3^3x_4^2x_5^2x_6^2x_7x_8^2 + x_2x_3x_4x_5^2x_6x_7x_8^2 \\
\deg(f_2) = 10: \quad f_2 &= x_1x_2^2x_4^2x_8x_9^2x_{10}^2 + x_2^2x_4x_5^2x_6x_7x_9x_{10}^2 + x_1^2x_2x_3x_5^2x_7^2x_9^2 + x_1x_3^2x_4^2x_7^2x_9^2 + x_1^2x_3x_4x_7^2x_8^2 \\
\deg(f_3) = 25: \quad f_3 &= 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3 \\
\deg(f_4) = 19: \quad f_4 &= 9x_1^2x_3x_4x_6^3x_7^2x_8x_{10}^4 + 17x_1^3x_2x_5^2x_6^2x_7x_8^3x_9^4x_{10}^3 + 17x_2^2x_3^4x_4^2x_7^4x_8^3x_9x_{10}^3 + 3x_1^3x_2^2x_6^3x_{10}^2 + 10x_1x_3x_5^2x_6^2x_7^4x_8^4
\end{aligned}
$$

	Thresholds			mod 11			mod 13		
	η, ζ	τ	κ, γ	$=$	$\neq$	!	$=$	$\neq$	!
f_1	2	2	6	28	2	70	28	0	72
Average black box probes: $=$				151.3928571			196.2142857		
Average black box probes: $=, \neq$				151.1666667			196.2142857		
f_2	2	2	6	8	1	91	26	0	74
Average black box probes: $=$				162.			164.3076923		
Average black box probes: $=, \neq$				161.6666667			164.3076923		
f_3	2	2	6	7	1	92	2	0	98
Average black box probes: $=$				167.2857143			167.		
Average black box probes: $=, \neq$				167.1250000			167.		
f_4	2	2	6	5	0	95	0	1	99
Average black box probes: $=$				170.					
Average black box probes: $=, \neq$				170.			180.		



The order of elements in mod 11



The order of elements in mod 13

Effects from the moduli.

The magnitude of modulus effects the randomization and hence the success rate in our Newton interpolation.

The number of elements of different orders in the modulus effects the success rate of our univariate Ben-Or/Tiwari algorithm.

Future Research

I hope to do at least 2 of the following:

Implement sparse shifts in Maple. Implement the sparse shifts proposed by Grigoriev and Lakshman in Maple.

Early termination on sparse shifts. Investigate whether the early termination can be implemented on the sparse shifts algorithm.

Probability analysis for thresholds. Do a better probability analysis for thresholds larger than 1, or the heuristic models for small moduli.

Very small coefficient fields. Note that for a small finite coefficient field, say $\mathbb{Z}_2$, one can switch to the coefficient domain $\mathbb{Z}_2[x_n]$, where x_n is the last variable. Investigate the proceeding modulo irreducible polynomials in $\mathbb{Z}_2[x_n]$.