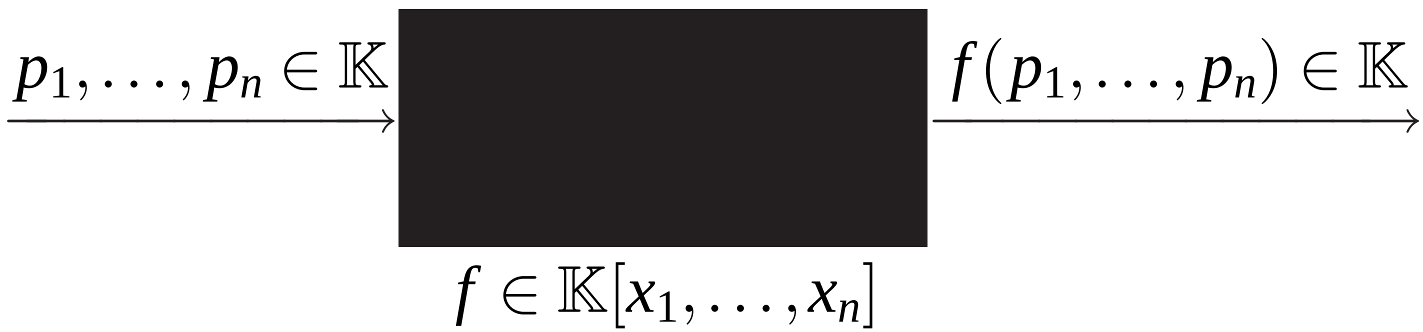
# FoxBox: A System for Manipulating Symbolic Objects in Black Box Representation

Erich Kaltofen
North Carolina State University
Raleigh, USA
kaltofen@math.ncsu.edu

Joint work with Angel Díaz
IBM T. J. Watson Research Center
Yorktown Heights, New York USA
aldiaz@us.ibm.com

# Black Box Model for Multivariate Polynomials




$\mathbb{K}$ an arbitrary field, e.g., rationals, reals, complexes

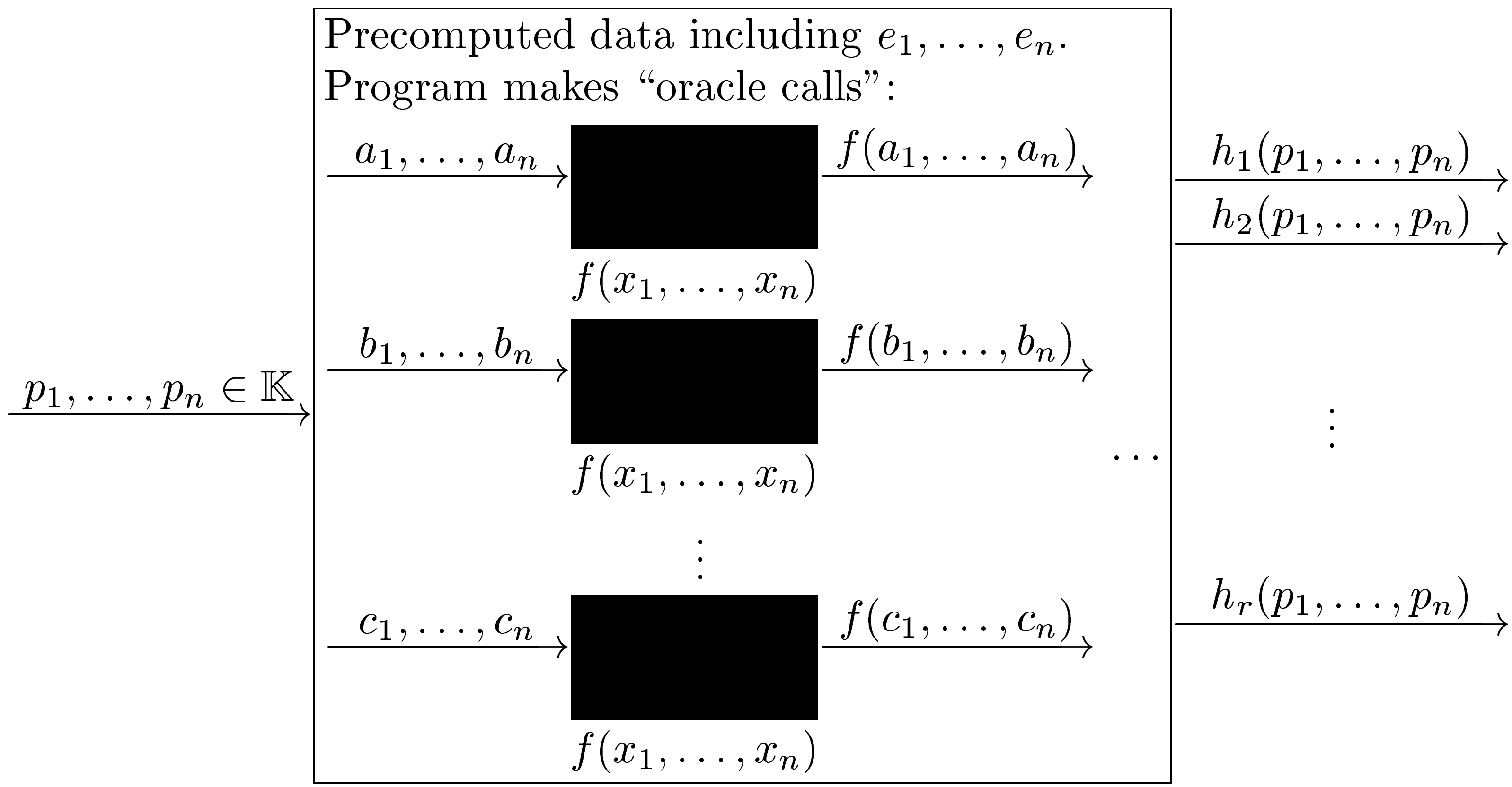
Perform polynmial algebra operations, e.g., factorization [K & Trager 88] with

$n^{O(1)}$ black box calls,
$n^{O(1)}$ arithmetic operations in $\mathbb{K}$ and
$n^{O(1)}$ randomly selected elements in $\mathbb{K}$

Given a black box


$f(x_1, \ldots, x_n) \in \mathbb{K}[x_1, \ldots, x_n]$
$\mathbb{K}$ a field of characteristic 0

K and TRAGER (1988) efficiently construct the following efficient program:

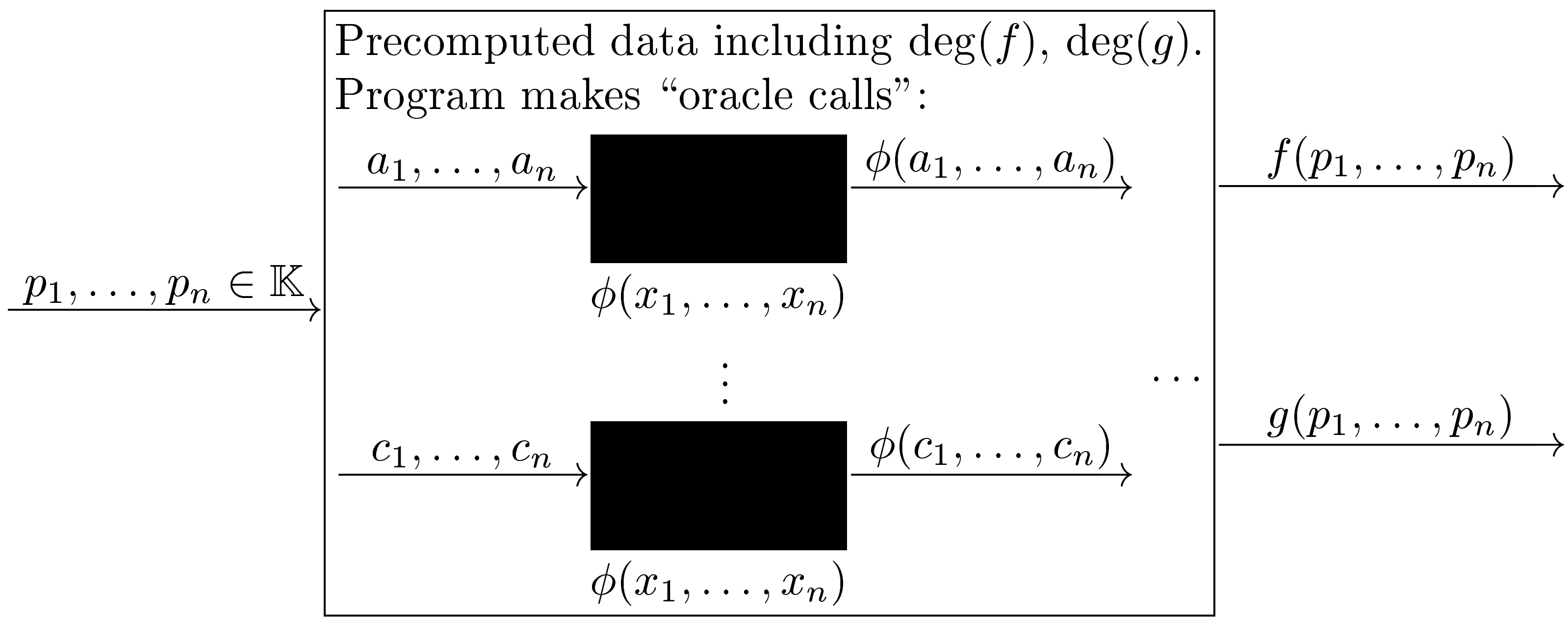



$$f(x_1, \ldots, x_n) = h_1(x_1, \ldots, x_n)^{e_1} \cdots h_r(x_1, \ldots, x_n)^{e_r}$$
$$h_i \in \mathbb{K}[x_1, \ldots, x_n] \text{ irreducible.}$$

Cauchy matrix: factorization of numerator

$$\det\left(\begin{bmatrix} \frac{1}{x_1+y_1} & \frac{1}{x_1+y_2} & \cdots & \frac{1}{x_1+y_n} \\ \frac{1}{x_2+y_1} & \frac{1}{x_2+y_2} & \cdots & \frac{1}{x_2+y_n} \\ \vdots & \vdots & & \vdots \\ \frac{1}{x_n+y_1} & \frac{1}{x_n+y_2} & \cdots & \frac{1}{x_n+y_n} \end{bmatrix}\right) = \frac{\prod\limits_{1\le i<j\le n} (x_j - x_i)(y_j - y_i)}{\prod\limits_{1\le i,j\le n} (x_i + y_j)}.$$

K and TRAGER (1988) efficiently construct the following efficient program:




$$\phi(x_1, \ldots, x_n) = \frac{f(x_1, \ldots, x_n)}{g(x_1, \ldots, x_n)}, f, g \in \mathbb{K}[x_1, \ldots, x_n], \mathrm{GCD}(f, g) = 1.$$

# Sparse Multivariate Interpolation

Given a black box


$f(x_1, \ldots, x_n) \in \mathbb{K}[x_1, \ldots, x_n]$

$\mathbb{K}$ a field of characteristic 0

compute by multiple evaluation of this black box the sparse representation of $f$

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{t} a_i x_1^{e_{i,1}} \cdots x_n^{e_{i,n}}, \quad a_i \neq 0$$

Several solutions that are polynomial-time in $n$ and $t$:

ZIPPEL (1979, 1988), BEN-OR, TIWARI (1988)
K, LAKSHMAN (1988), GRIGORIEV, KARPINSKI, SINGER (1988)
MANSOUR (1992)

# Sparsity with non-standard basis

In place of $x^e$ use

| | |
|---|---|
| $(x-a)^e$ | shifted basis |
| $x(x+1)\cdots(x+e-1)$ | Pochhammer basis |
| $T_e(x)$ | Chebyshev basis |

Solutions (not all polynomial-time):

LAKSHMAN, SAUNDERS (1992, 1994): Chebyshev, Pochh., shifted
GRIGORIEV, KARPINSKI (1993): shifted
GRIGORIEV, LAKSHMAN (1995): shifted

## The FoxBox System [Díaz & Kaltofen 1998

- Base arithmetics
- Black box objects
- Common black box objects
- Black box algorithms
- Extended domain black box objects
- Homomorphic maps
- Parallel black boxes

# Parallel black boxes

Once constructed, a black box algorithm utilizes a small amount of precomputed static information for evaluation.

- The parallel black box interface adds three member functions for administering remote evaluation

- An initialization phase transmits static information to each processor allowing for subsequent probes

- FoxBox provides an MPI compliant implementation of the parallel black boxes interface

Running example: determinant of symmetric Toeplitz matrix

$$\det\left(\begin{bmatrix} a_0 & a_1 & \dots & a_{n-2} & a_{n-1} \\ a_1 & a_0 & \dots & a_{n-3} & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \dots & a_0 & a_1 \\ a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \end{bmatrix}\right)$$

$$= F_1(a_0, \dots, a_{n-1}) \cdot F_2(a_0, \dots, a_{n-1}).$$

over the integers.

```
> readlib(showtime):
> showtime():
O1 := T := linalg[toeplitz]([a,b,c,d,e,f]):
time   0.03   words   7701
O2 := factor(linalg[det](T));
```

$$
\begin{aligned}
&-(2\,d\,c\,a-2\,b\,c\,e+2\,c^2\,a-a^3-d\,a^2+2\,d^2\,c+d^2\,a+b^3+2\,a\,b\,c-2\,c^2\,b\\
&\quad+d^3+2\,a\,b^2-2\,d\,c\,b-2\,c\,b^2-2\,e\,c^2+2\,e\,b^2+2\,f\,c\,b+2\,b\,a\,e\\
&\quad+b^2\,f+c^2\,f+b\,e^2-b\,a^2-f\,d\,b-f\,d\,a-f\,a^2-f\,b\,a+e^2\,a-2\,d\,b^2\\
&\quad+d\,c^2-2\,d\,e\,b-2\,d\,e\,c-d\,b\,a)(2\,d\,c\,a-2\,b\,c\,e-2\,c^2\,a+a^3\\
&\quad-d\,a^2-2\,d^2\,c-d^2\,a+b^3+2\,a\,b\,c-2\,c^2\,b+d^3-2\,a\,b^2+2\,d\,c\,b\\
&\quad+2\,c\,b^2+2\,e\,c^2-2\,e\,b^2-2\,f\,c\,b+2\,b\,a\,e+b^2\,f+c^2\,f+b\,e^2-b\,a^2\\
&\quad-f\,d\,b+f\,d\,a-f\,a^2+f\,b\,a-e^2\,a-2\,d\,b^2+d\,c^2+2\,d\,e\,b-2\,d\,e\,c\\
&\quad+d\,b\,a)
\end{aligned}
$$

```
time   27.30   words   857700
```

Example: a FoxBox common object and factor box

```
// initialize our SACLIB and NTL wrapper/adaptors
Word Stack; int i;
SaclibInitEnv( 1000000, Stack ); ShoupInitEnv( &MPCard );
```

```
// construct a symmetric Toeplitz determinant
// from which we create a factor box
typedef BlackBoxSymToeDet< SaclibQ, SaclibQX >  BBSymToeDetQ;
typedef BlackBoxFactors< SaclibQ, SaclibQX,
                         BBSymToeDetQ > BBFactorsQ;

BBSymToeDetQ SymToeDetQ( N, DegDet );
BBFactorsQ   FactorsQ( SymToeDetQ, Prob, Seed, &MPCard );
```

# Example: a FoxBox homomorphic image

```
// map the factors black box to NTL's modular arithmetic
typedef BlackBoxSymToeDet< ShoupZP, ShoupZPX > BBSymToeDetZP;

typedef BlackBoxFactorsHMap< SaclibQ, SaclibQX,
                             ShoupZP, ShoupZPX,
                             BBSymToeDetQ, BBSymToeDetZP,
                             SaclibQShoupZP > BBFactorsQMapZP;


SaclibQShoupZP   h;
BBSymToeDetZP    SymToeDetZP( N, DegDet );
BBFactorsQMapZP  FactorsZP( SymToeDetZP, FactorsQ, h );

SaclibCleanUpEnv(); // we no longer need rational arithmetic
```

Example: FoxBox sparse interpolation

```
// interpolate the first factor
typedef BlackBoxSelector< ShoupZP, BBFactorsQMapZP > BBFactorZP;

BBFactorZP FirstFactorZP( FactorsZP, 0 );

SparseInterp( FirstFactorZP, Vars, Degs,
              DegDet, &MPCard,
              AnsDegs, AnsMons,
              ShoupZPXElm,
              ProbName, IsRestart, NumProc  );

ShoupCleanUpEnv();
```

# Factorization challenge: construction

| $N$ | CPU Time | $N$ | CPU Time |
|---|---|---|---|
| 11 | $0^h 02'$ | 16 | $0^h 43'$ |
| 12 | $0^h 05'$ | 17 | $1^h 05'$ |
| 13 | $0^h 09'$ | 18 | $1^h 42'$ |
| 14 | $0^h 16'$ | 19 | $2^h 30'$ |
| 15 | $0^h 26'$ | 20 | $3^h 42'$ |

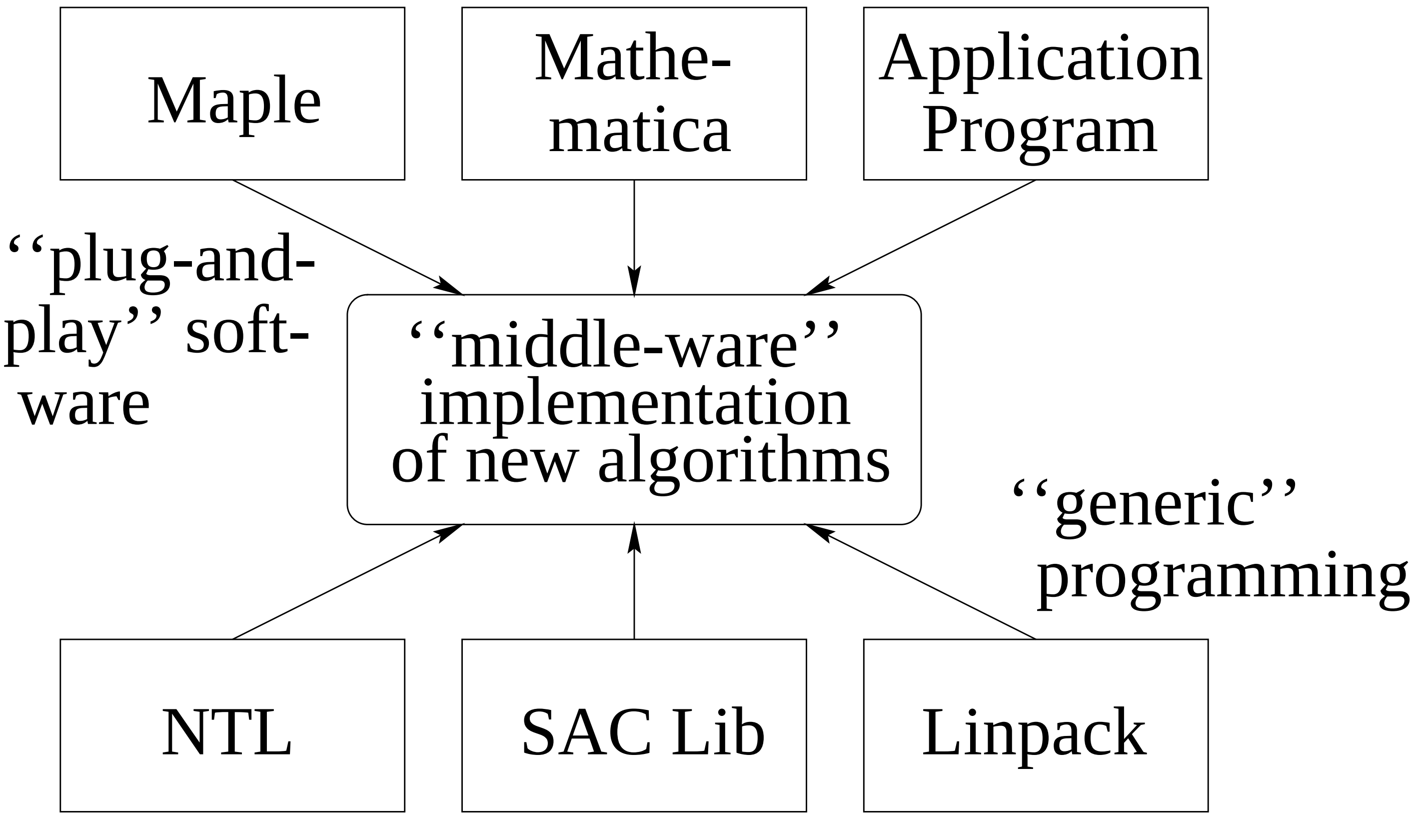
Total CPU times (hours$^h$minutes$'$) required to construct a factors black box (over $\mathbb{Q}$) that can evaluate both irreducible factors of the determinant of a symmetric Toeplitz matrix. The processor is a Sun Ultra 1/170 (128MB), Solaris 2.5.

# Factorization challenge: sparse conversion

| $N$ | CPU Time | Degree | # Terms |
|---|---|---|---|
| 10 | $1^h20'$ | 5 | 931 |
| 11 | $1^h34'$ | 5 | 847 |
| 12 | $10^h14'$ | 6 | 5577 |
| 13 | $15^h24'$ | 6 | 4982 |

CPU times (hours$^h$minutes$'$) to retrieve the distributed representation of a factor from the factors black box of a symmetric Toeplitz determinant black box. Construction is over $\mathbb{Q}$ evaluation is in $\mathrm{GF}(10^8+7)$ for $N = 10, 11$, and 12 (Pentium 133, Linux 2.0) and $\mathrm{GF}(2^{30}-35)$ for $N = 13$ (Sun Ultra 2 168MHz, Solaris 2.4).

Plug-And-Play Components




Problem solving environ’s: end-user can easily custom-make symbolic software

Example: Maple black box factorization

```
> SymToeQ  := BlackBoxSymToe( BBNET_Q, 4, -1, 1.0 ):

> SymToeZP := BlackBoxSymToe( BBNET_ZP, 4, -1, 1.0 ):

> FactorsQ := BlackBoxFactors( BBNET_Q, SymToeQ, Mod, 1.0,
                              Seed ):

> FactorsZP := BlackBoxHomomorphicMap( BBNET_FACS, FactorsQ,
                                      SymToeZP ):

> FactorZP  := BlackBoxSelectValue( BBNET_ZP, FactorsZP, 0 ):

> FB1 := SparseConversion( BBNET_ZP, FactorZP,
             [ x1, x2, x3, x4 ], [ 4, 4, 4, 4 ], 4, Mod );
```

# Software Design Issues

## Plug-and-play

- Standard representation for transfer: MP, OpenMath, MathML
- Byte code for constructing objects vs. parse trees
- Visual programming environments for composition

## Generic Programming

- Common object interface (wrapper classes),
  e.g., `K::random_generator(500)`
- Storage management vs. garbage collection
- Algorithmic shortcuts into the basic modules