Computer algebra in the new century
The road ahead

Erich Kaltofen
North Carolina State University
www.kaltofen.net

KUNGL. VETENSKAPSAKADEMIEN
THE ROYAL SWEDISH ACADEMY OF SCIENCES

Information Department, PO Box 50005, SE-104 05 Stockholm, Sweden, webbsite: www.kva.se
Tel: +46-8-673 95 95, Fax +46-8-15 56 70, e-mail: info@kva.se

THE NOBEL PRIZE IN PHYSICS 1999

PRESS RELEASE 12 OCTOBER 1999
The Prize I Further reading I The laureates

The Royal Swedish Academy of Sciences has awarded
the 1999 Nobel Prize in Physics
jointly to

Professor **Gerardus 't Hooft**, University of Utrecht, Utrecht, the Netherlands, and
Professor Emeritus **Martinus J.G. Veltman**, University of Michigan, USA, resident in Bilthoven, the Netherlands.

The two researchers are being awarded the Nobel Prize for having placed particle physics theory on a firmer mathematical foundation. **...**

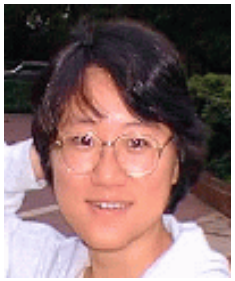
The Academy's citation:
"for elucidating the quantum structure of electroweak interactions in physics."

...
One person who had not given up hope of being able to renormalize non-abelian gauge theories was **Martinus J.G.Veltman**. At the end of the 1960s he was a newly appointed professor at the University of Utrecht. Veltman had developed the *Schoonschip* computer program which, using symbols, performed algebraic simplifications of the complicated expressions that all quantum field theories result in when quantitative calculations are performed. Twenty years earlier, Feynman had indeed systematised the problem of calculation and introduced *Feynman diagrams* that were rapidly accepted by researchers. But at that time there were no computers. Veltman believed firmly in the possibility of finding a way of renormalizing the theory and his computer program was the cornerstone of the comprehensive work of testing different ideas.

Overview

1. Faster algorithms:
   counting bit operations vs. counting arithmetic operations

2. Imprecise inputs
   With PhD student Markus Hitz 

3. Uncertain results
   With PhD student Wen-shin Lee 

1. Linear Algebra

Strassen's [1969] $O(n^{2.81})$ matrix multiplication algorithm

$$
\begin{array}{l|l}
m_1 \leftarrow (a_{1,2} - a_{2,2})(b_{2,1} - b_{2,2}) & \\
m_2 \leftarrow (a_{1,1} + a_{2,2})(b_{1,1} + b_{2,2}) & \\
m_3 \leftarrow (a_{1,1} - a_{2,1})(b_{1,1} + b_{1,2}) & \\
m_4 \leftarrow (a_{1,1} + a_{1,2})b_{2,2}) & a_{1,1}b_{1,1} + a_{1,2}b_{2,1} = m_1 + m_2 - m_4 + m_6 \\
m_5 \leftarrow a_{1,1}(b_{1,2} - b_{2,2}) & a_{1,1}b_{1,2} + a_{1,2}b_{2,2} = m_4 + m_5 \\
m_6 \leftarrow a_{2,2}(b_{2,1} - b_{1,1}) & a_{2,1}b_{1,1} + a_{2,2}b_{2,1} = m_6 + m_7 \\
m_7 \leftarrow (a_{2,1} + a_{2,2})b_{1,1}) & a_{2,1}b_{1,2} + a_{2,2}b_{2,2} = m_2 - m_3 + m_5 - m_7
\end{array}
$$

Problems reducible to matrix multiplication:

linear system solving [Bunch and Hopcroft 1974],...

Coppersmith and Winograd [1990]: $O(n^{2.38})$

Life after Strassen: black box linear algebra

The black box model of a matrix

$A \in \mathbb{K}^{n \times n}$ singular

$\mathbb{K}$ an arbitrary, e.g., finite field

Perform linear algebra operations, e.g., $A^{-1}b$ [Wiedemann 86] with

$O(n)$	black box calls and
$n^2(\log n)^{O(1)}$	arithmetic operations in $\mathbb{K}$ and
$O(n)$	intermediate storage for field elements

Flurry of recent results

Lambert [96], Teitelbaum [98], Eberly & Kaltofen [97]	relationship of Wiedemann and Lanczos approach
Villard [97]	analysis of *block* Wiedemann algorithm
Giesbrecht [97] and Mulders & Storjohann [99]	computation of integral solutions
Giesbrecht, Lobo & Saunders [98]	certificates for inconsistency
Chen, Eberly, Kaltofen, Saunders, Villard & Turner [2K]	butterfly network, sparse and diagonal preconditioners
Villard [2K]	characteristic polynomial

Diophantine solutions
by Giesbrecht:
Find several rational solutions.

$$A(\tfrac{1}{2}x^{[1]}) = b, \quad x^{[1]} \in \mathbb{Z}^n$$
$$A(\tfrac{1}{3}x^{[2]}) = b, \quad x^{[2]} \in \mathbb{Z}^n$$
$$\gcd(2,3) = 1 = 2 \cdot 2 - 1 \cdot 3$$
$$A(2x^{[1]} - x^{[2]}) = 4b - 3b = b$$

LINSOLVE0: *Given blackbox A, compute $w \neq 0$ such that $Aw = 0$.*

NONSINGULAR$\leq$LINSOLVE0: For $Ax = b$ solve $\left[A \mid b\right] w = 0$ and compute $x = \frac{1}{w_{n+1}} \begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix}$.

Used in sieve-based integer factoring algorithms.

Harder (?) problem

LINSOLVE1: *Given blackbox A (possibly singular) and b, compute x such that $Ax = b$.*

Random sampling in the nullspace is equivalent to LINSOLVE1: select a random vector y and solve $Ax = b$ for $b = Ay$.

Life after Strassen: bit complexity

Linear system solving $x = A^{-1}b$ where $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^n$:

With Strassen [McClellan 1973]:

Step 1: For prime numbers $p_1, \ldots p_k$ Do
Solve $Ax^{[j]} \equiv b \pmod{p_j}$ where $x^{[j]} \in \mathbb{Z}/(p_j)$

Step 2: Chinese remainder $x^{[1]}, \ldots, x^{[k]}$ to $A\overline{x} \equiv b \pmod{p_1 \cdots p_k}$

Step 3: Recover denominators of x_i by continued fractions of $\dfrac{\overline{x_i}}{p_1 \cdots p_k}$.

Length of integers: $k = (n \max\{\log \|A\|, \log \|b\|\})^{1+o(1)}$

Bit complexity: $n^{3.38} \max\{\log \|A\|, \log \|b\|\}^{1+o(1)}$

With Hensel  lifting [Moenck and Carter 1979]:

Step 1: For $j = 0, 1, \ldots, k$ and a prime p Do

Compute $\vec{x}^{[j]} = x^{[0]} + px^{[1]} + \cdots + p^j x^{[j]} \equiv x \pmod{p^{j+1}}$

1.a. $b^{[j]} = \dfrac{b - A\vec{x}^{[j-1]}}{p^j} = \dfrac{b - (A\vec{x}^{[j-2]} + Ap^{j-1}x^{[j-1]})}{p^j}$

1.b. $x^{[j]} \equiv A^{-1}b^{[j]} \pmod{p}$ reusing $A^{-1} \bmod p$

Step 3: Recover denominators of x_i by continued fractions of $\dfrac{\vec{x}_i^{[k]}}{p^k}$.

With classical matrix arithmetic:

Bit complexity of 1.a: $n(n\max\{\log\|A\|, \|b\|\})^{1+o(1)} + n^2(\log\|A\|)^{1+o(1)}$

Total bit complexity: $(n^3 \max\{\log\|A\|, \log\|b\|\})^{1+o(1)}$

Bit complexity of the determinant

Wiedemann's [1986] determinant algorithm
For $u, v \in \mathbb{K}^n$ and $A \in \mathbb{K}^{n \times n}$ consider the sequence of field elements

$$a_0 = u^T v,\ a_1 = u^T A v,\ a_2 = u^T A^2 v,\ a_3 = u^T A^3 v,\ \ldots$$

The minimal polynomial of A linearly generates $\{a_i\}_{i=0,1,\ldots}$.

By the Berlekamp/Massey [1967] algorithm we can compute in $n^{1+o(1)}$ arithmetic operations a minimal linear generator for $\{a_i\}_{i=0,1,\ldots}$.

Wiedemann randomly perturbs A and chooses random u and v; then

$$\det(\lambda I - A) = \text{minimal recurrence polynomial of } \{a_i\}_{i=0,1,\ldots}.$$

Detail of algorithm
[exactly like my division-free determinant algorithm ISSAC 92]

For $i = 0, 1, \ldots, 2n-1$ Do Compute the $a_i = u^T A^i v$;

Done by baby steps/giant steps: let $r = \lceil \sqrt{2n} \rceil$ and $s = \lceil 2n/r \rceil$.

Substep 1. For $j = 1, 2, \ldots, r-1$ Do $v^{[j]} \leftarrow A^j v$;

Substep 2. $Z \leftarrow A^r$;
[$O(n^3)$ operations; integer length $(\sqrt{n}\ \log \|A\|)^{1+o(1)}$]

Substep 3. For $k = 1, 2, \ldots, s$ Do ${u^{[k]}}^T \leftarrow u^T Z^k$;
[$O(n^{2.5})$ operations; integer length $(n\ \log \|A\|)^{1+o(1)}$]

Substep 4. For $j = 0, 1, \ldots, r-1$ Do
For $k = 0, 1, \ldots, s$ Do $a_{kr+j} \leftarrow \langle u^{[k]}, v^{[j]} \rangle$.

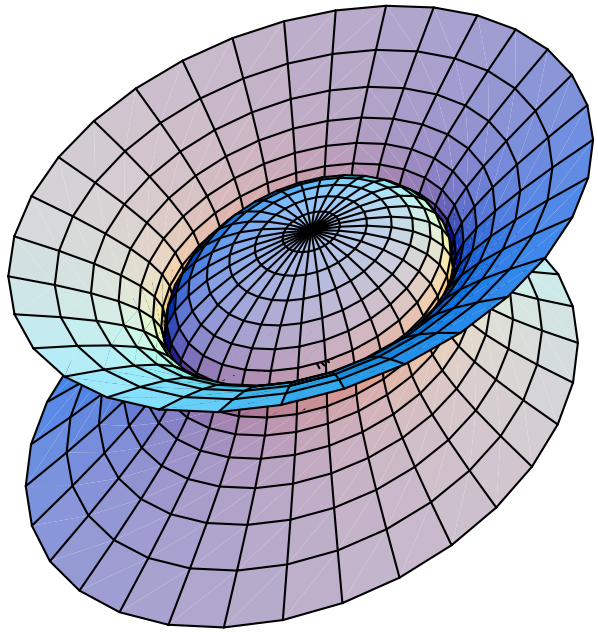
Using fast rectangular matrix multiplication: $O(n^{3.064} \log \|A\|)$

Problem 1

Improve the bit complexity of algorithms for the determinant, resultant, linear system solution, over the integers.

2. Factorization of nearby polynomials over the complex numbers

$$81x^4+16y^4-648z^4+72x^2y^2-648x^2-288y^2+1296=0$$



$$(9x^2+4y^2+18\sqrt{2}z^2-36)(9x^2+4y^2-18\sqrt{2}z^2-36)=0$$

$$81x^4+16y^4-648.003z^4+72x^2y^2+.002x^2z^2+.001y^2z^2$$
$$-648x^2-288y^2-.007z^2+1296=0$$

Problem 2 *[Kaltofen LATIN'92]*
Given is a polynomial $f(x,y) \in \mathbb{Q}[x,y]$ *and* $\varepsilon \in \mathbb{Q}$.

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

Theorem *[Hitz, Kaltofen, Lakshman ISSAC'99]*
We can compute in polynomial time in the degree and coefficient size if there is a $\tilde{f}(x,y) \in \mathbb{C}[x,y]$ *with a factor of a* ***constant*** *degree and* $\|f - \tilde{f}\|_2 \leq \varepsilon$.

Numerical algorithms

Conclusion on my exact algorithm [JSC 1985]:

"*D. Izraelevitz at Massachusetts Institute of Technology has already implemented a version of algorithm 1 using complex floating point arithmetic. Early experiments indicate that the linear systems computed in step (L) tend to be* ***numerically ill-conditioned.*** *How to overcome this numerical problem is an important question which we will investigate.*"

Galligo and Watt [ISSAC'97]: substitute variables by generic linear forms; then certain coefficients in true factors must vanish.

Stetter, Haung, Wu and Zhi [ISSAC'2K]: Hensel lift factor combinations numerically and eliminate extraneous factors early

Univariate Problem: Given $f \in \mathbb{C}[z]$ and $\alpha \in \mathbb{C}$.
Find $\tilde{f} \in \mathbb{C}[z]$,such that

$$\tilde{f}(\alpha) = 0\,, \quad \text{and} \quad \|f - \tilde{f}\| = \min\,.$$

Let

$$\begin{aligned} f(z) &= a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \\ \tilde{f}(z) &= (z-\alpha)\left(u_{n-1} z^{n-1} + u_{n-2} z^{n-2} + \cdots + u_0\right) \\ &= u_{n-1} z^n + (u_{n-2} - \alpha) z^{n-1} + \cdots + (u_0 - \alpha u_1) z - \alpha u_0 \end{aligned}$$

In terms of linear algebra:

$$\|f - \tilde{f}\| = \min_{\mathbf{u} \in \mathbb{C}^n} \left\| \underbrace{\begin{bmatrix} -\alpha & & & 0 \\ 1 & -\alpha & & \\ & \ddots & \ddots & \\ 0 & & 1 & -\alpha \\ & & & 1 \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix}}_{\mathbf{u}} - \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix}}_{\mathbf{b}} \right\| \qquad (1)$$

(1) is an over-determined linear system of equations:

Linear program, if $\|\cdot\|$ is the $\begin{cases} \infty\text{-norm, or} \\ 1\text{-norm} \end{cases}$

Least squares problem, if $\|\cdot\|$ is the 2-norm (Euclidean).

Solutions for the 2-norm in closed form:

$$N_{min}(\alpha) = \|f - \tilde{f}\|^2 = \frac{\overline{f(\alpha)}\, f(\alpha)}{\sum_{k=0}^{n} (\overline{\alpha}\alpha)^k}, \qquad f_j - \tilde{f}_j = \frac{(\overline{\alpha})^j f(\alpha)}{\sum_{k=0}^{n} (\overline{\alpha}\alpha)^k}$$

(also derived in Corless et al. [ISSAC'95] via SVD)

Constraining a Root Locus to a Curve

Let Γ be a piecewise smooth curve with finitely many segments, each having a parametrization $\gamma_k(t)$ in a single real parameter t.

For a given polynomial $f \in \mathbb{C}[z]$, we want to find a minimally perturbed polynomial $\tilde{f} \in \mathbb{C}[z]$ that has (at least) one root on Γ.

Parametric Minimization

We substitute the parametrization $\gamma_k(t)$ for the indeterminate α in $N_{min}(\alpha)$. The resulting expression is a function in $t \in \mathbb{R}$.

It attains its minima at its *stationary* points. We have to compute the *real* roots of the derivative.

The derivative of the norm-expression is determined *symbolically*, but the roots can be computed numerically.

Bivariate factorization

Given $f = \sum f_{i,j} x^i y^j \in \mathbb{C}[x,y]$ **absolute irreducible**, find

$$\tilde{f} = (c_0 + c_1 x + c_2 y) u(x,y) \in \mathbb{C}[x,y], \quad \deg(\tilde{f}) \leq \deg(f),$$

such that $\| f - \tilde{f} \|_2$ is minimal.

("nearest polynomial with a linear factor").

Approach: minimize parametric least square solution in the real and imaginary parts of the $c_i = \alpha_i + \beta_i \mathbf{i}$.
$\rightarrow$ must minimize least squares solution with 6 parameters.
$\rightarrow$ yields polynomial system with a **fixed number of variables**, hence polynomial time.

An ∞-norm example: $x^2+1 = 1\cdot x^2 + 0\cdot x + 1$

$$\min_{\tilde{a}_2,\tilde{a}_1,\tilde{a}_0 \in \mathbb{R} \text{ such that } \exists \alpha \in \mathbb{R}:\ \tilde{a}_2\alpha^2+\tilde{a}_1\alpha+\tilde{a}_0 = 0} \left(\max\{|1-\tilde{a}_2|, |0-\tilde{a}_1|, |1-\tilde{a}_0|\right)$$

$$=?$$

Special case: nearest polynomial with root α :

$$\delta(\alpha) = \min_{\mathbf{u}\in\mathbb{R}^n} \|\mathbf{Pu} - \mathbf{b}\|_\infty = \left| \frac{\sum_{i=0}^{n} \lambda_i a_i}{\sum_{i=0}^{n} |\lambda_i|} \right| = \left| \frac{f(\alpha)}{\sum_{i=0}^{n} |\alpha^i|} \right| . \qquad (2)$$

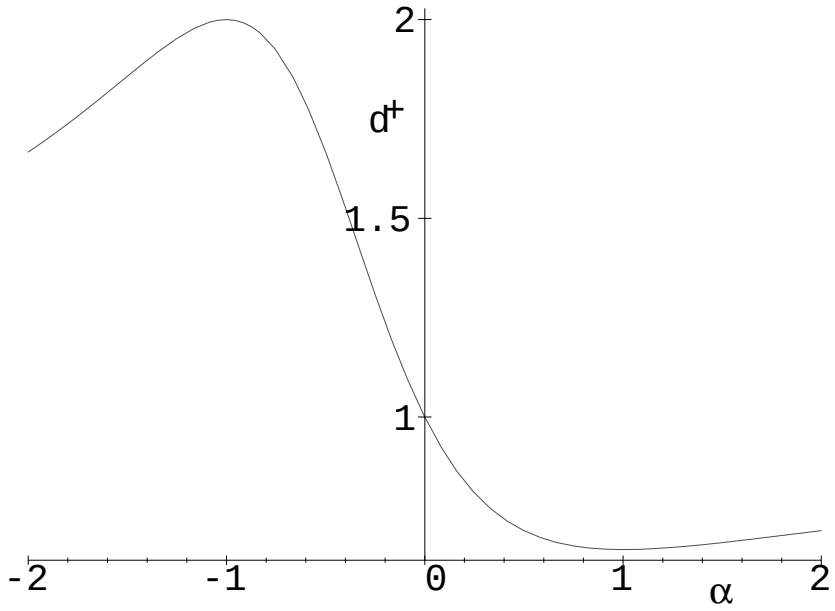
(also derived by Manocha & Demmel [1995])

Stiefel's theorem also gives algorithm for finding $\mathbf{u}$.

Parametric α : must minimize rational function (2).

Generalization to l^p-norm, where $1 \le p \le \infty$ (Hitz 1999):

$$\delta(\alpha) = \frac{|f(\alpha)|}{\left(\sum_{k=0}^{n} |\alpha^k|^q\right)^{1/q}}, \quad \frac{1}{q} + \frac{1}{p} = 1, \quad \text{and} \quad \frac{1}{\infty} = 0$$



$$f(x) = x^2 + 1,\ \tilde{f}(x) = \tfrac{1}{3}x^2 - \tfrac{2}{3}x + \tfrac{1}{3} = \tfrac{1}{3}(x-1)^2,\ \delta = \tfrac{2}{3}.$$

Sensitivity analysis: component-wise nearest singular matrix

Given are $2n^2$ rational numbers $\underline{a}_{i,j}, \overline{a}_{i,j}$.
Let A be the *interval* matrix

$$A = \left\{ \begin{bmatrix} a_{1,1} & \dots & a_{n,n} \\ \vdots & & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{bmatrix} \mid \underline{a}_{i,j} \leq a_{i,j} \leq \overline{a}_{i,j} \text{ for all } 1 \leq i, j \leq n \right\}.$$

Does A contain a singular matrix?
This problem is *NP-complete* (Poljak & Rohn 1990).

3. Will our systems guarantee their answers?

Maple 6 allows calls to NAG numeric library routines

Basic polynomial algorithms with floating point coefficients are under development

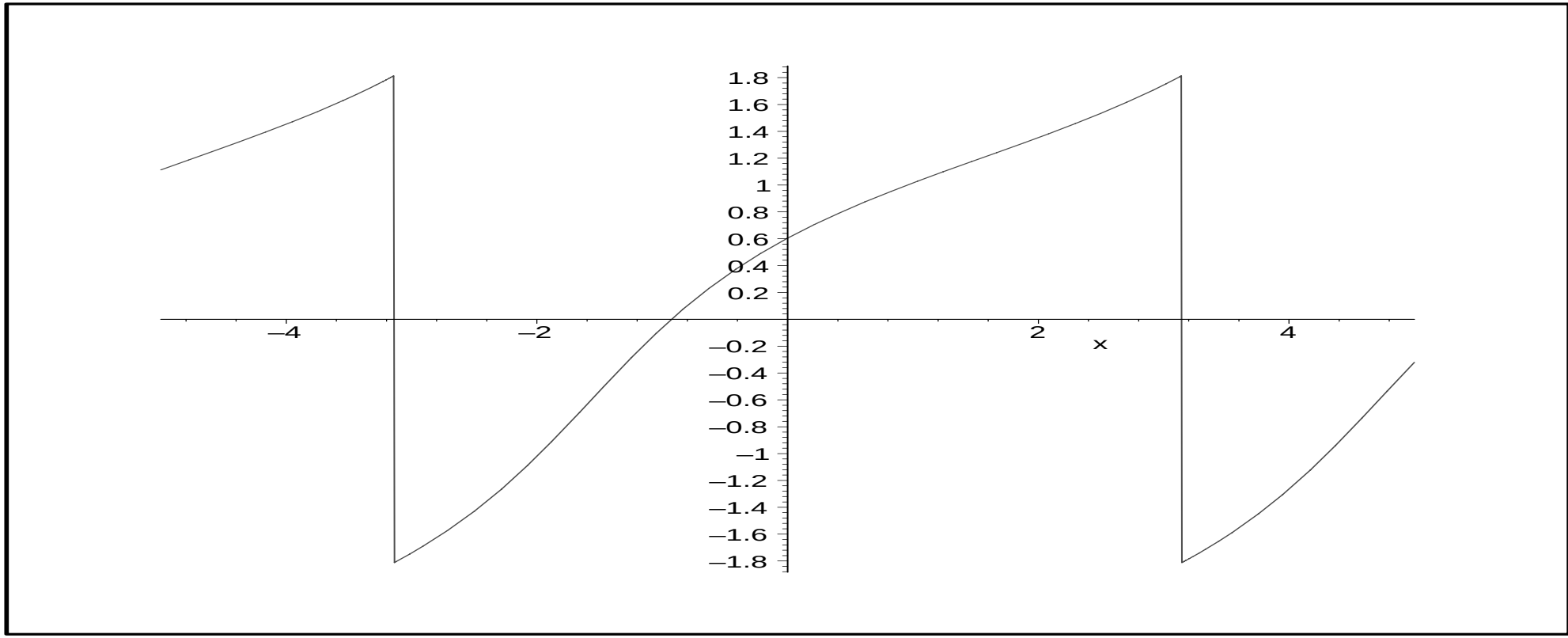
```
> # Example by Corless and Jeffrey
> f := 1/(sin(x) + 2);
```

$$f := \frac{1}{\sin(x)+2}$$

```
> g := int(f, x);
```

$$g := \frac{2}{3}\sqrt{3}\arctan\left(\frac{1}{3}\left(2\tan\left(\frac{1}{2}x\right)+1\right)\sqrt{3}\right)$$

```
> plot(g, x=-5..5);
```



Early termination strategies

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.

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

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

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

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

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

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.

```
> read `../../initpkg.mpl`:
> with(protobox);
```

[*BM_step_mod*, *HybridInterp*, *NewtonInterp_step*, *bbpoly_mod*, *check_same*, *eval_mon_mod*, *eval_polyseq*, *eval_tmpprunelist_mod*, *find_max*, *find_true*, *heval_plist_mod*, *heval_pnt_mod*, *list_to_poly*, *prune*, *raising_pnts_mod*, *recover*, *relocate_c*, *relocate_shift_c*, *rev*, *rm_element*, *slice*, *spoly_to_slist*, *tmpprune*, *tmpprunelist_to_plist*, *vansolve_kl_mod*]

```
> f_3:=9*x[2]^3*x[3]^3*x[5]^2*x[6]^2*x[8]^3*x[9]^3 +
> 9*x[1]^3*x[2]^2*x[3]^3*x[5]^2*x[7]^2*x[8]^2*x[9]^3 +
> x[1]^4*x[3]^4*x[4]^2*x[5]^4*x[6]^4*x[7]*x[8]^5*x[9] +
> 10*x[1]^4*x[2]*x[3]^4*x[4]^4*x[5]^4*x[7]*x[8]^3*x[9] +
> 12*x[2]^3*x[4]^3*x[6]^3*x[7]^2*x[8]^3;
```

$$f\_3 := 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3$$

```
> bb:=bbpoly_mod(f_3,[x[1],x[2],x[3],x[4],x[5],x[6],x[7],x[8],x[9],x[10]
> ]);
```

```
bb := proc(pntsnf_i, modulusnf_i)
  local polynf, inf, numvarsnf, varlistnf;
    polynf := 9*x2^3*x3^3*x5^2*x6^2*x8^3*x9^3
      +9*x1^3*x2^2*x3^3*x5^2*x7^2*x8^2*x9^3
      +x1^4*x3^4*x4^2*x5^4*x6^4*x7*x8^5*x9
      +10*x1^4*x2*x3^4*x4^4*x5^4*x7*x8^3*x9
      +12*x2^3*x4^3*x6^3*x7^2*x8^3;
    numvarsnf := 10;
    varlistnf := [x1, x2, x3, x4, x5, x6, x7, x8, x9, x10];
    polynf := Eval(polynf, {seq(varlistnf[inf] = pntsnf_i[inf], inf = 1..numvarsnf)})
      mod modulusnf_i;
    RETURN(polynf)
  end
```

```
> NBT:=3; posttest:=2; rndmap:=4;
```

$NBT := 3$

$posttest := 2$

$rndmap := 4$

```
> m:=41; correct:=0; incorrect:=0; error:=0; bbcnt_all:=0;
```

$m := 41$

$correct := 0$

$incorrect := 0$

error := 0

bbcnt_all := 0

```
> bbcnt_correct:=0; num_test:=100;
```

bbcnt_correct := 0

num_test := 100

```
> for i from 1 to num_test do print(i,'-th');
> result_poly:=traperror(HybridInterp(bb,
> [x[1],x[2],x[3],x[4],x[5],x[6],x[7],x[8],x[9],x[10]],100,m,'N_thresh'=
> NBT, 'BM_thresh'=NBT, 'mapmon_thresh'=rndmap, 'rndrep_thresh'=rndmap,
> 'posttest_thresh'=posttest,'print_bbcnt'='num_bbprobe' ));
> if lasterror='lasterror' then
> if 0 = modp(expand(result_poly-f_3),m) then
> correct := correct + 1;
> bbcnt_all := bbcnt_all + num_bbprobe;
> bbcnt_correct := bbcnt_correct + num_bbprobe;
> else
> incorrect := incorrect + 1;
> bbcnt_all := bbcnt_all + num_bbprobe;
> fi;
> else
> error := error + 1;
> fi;
> od; print(`correct=`,correct); print(`incorrect=`,incorrect);
```

$1, -th$

Warning : range of random number generator or the modulus might not be enough

$$result\_poly := 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3$$

$2, -th$

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non – zero term*

$3, -th$

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non – zero term*

$4, -th$

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *In Zippel algorithm* : *different terms map to the same value*, 4, *times*

$5, -th$

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non – zero term*

$6, -th$

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non – zero term*

7, *– th*

Warning : *range of random number generator or the modulus might not be enough*

$$result\_poly := 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3$$

8, *– th*

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non – zero term*

9, *– th*

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non – zero term*

10, *– th*

Warning : *range of random number generator or the modulus might not be enough*

$$result\_poly := 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3$$

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non* − *zero term*

$99, -th$

Warning : *range of random number generator or the modulus might not be enough*

$$result\_poly := 9x_2^3x_3^3x_5^2x_6^2x_8^3x_9^3 + 9x_1^3x_2^2x_3^3x_5^2x_7^2x_8^2x_9^3 + x_1^4x_3^4x_4^2x_5^4x_6^4x_7x_8^5x_9 + 10x_1^4x_2x_3^4x_4^4x_5^4x_7x_8^3x_9 + 12x_2^3x_4^3x_6^3x_7^2x_8^3$$

$100, -th$

Warning : *range of random number generator or the modulus might not be enough*

result_poly := *Interpolation Failure* : *dropped a non* − *zero term*

correct =, 26

incorrect =, 0

```
> print(‘error=‘,error); if not expand(error-num_test)=0 then if
> not(correct=0) then print(‘average bbcnt for correct results=‘,
> evalf(bbcnt_correct/correct)); fi; print (‘overall average bbcnt=‘,
> evalf(bbcnt_all/(incorrect+correct))); fi;
```

error =, 74

average bbcnt for correct results =, 226.0384615

overall average bbcnt =, 226.0384615

Success and failure under different moduli and thresholds

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

	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

Problem 3

Provide reasonable correctness specifications for our systems in the presence of floating point numbers, randomizations, and multivalued functions.