An algorithmic approach to solving polynomial equations assocciated with quantum circuits

Vladimir Gerdt and Mikhail Zinin

Laboratory of Information Technologies
Joint Institute for Nuclear Research
141980, Dubna
Russia

The 4th International Workshop
"Quantum Physics and Communication" ,
15th October 2007, Dubna, Russia

Contents

Main Question and Notations

Constructing unitary matrices for quantum circuit built from the Hadamard and Toffoli gates is reduced to counting a number of solutions of multivariate polynomial systems associated with the circuits.

Main Question:

How to count the number of roots for multivariate polynomial systems algorithmically and efficiently?

Notations:

$\mathbb{X} = \{x_1, \ldots, x_n\}$ is the set of polynomial variables.
$R = \mathbb{K}[\mathbb{X}]$ is a polynomial ring over field $\mathbb{K}$ of characteristic 0.
$R' = \mathbb{F}_2[\mathbb{X}]$ is a polynomial ring over field $\mathbb{F}_2 = \{0, 1\}$.
$\tilde{R} = \mathbb{F}_2[\{x_1, \ldots, x_n\} \in \mathbb{F}_2^n]$ is a polynomial ring over field $\mathbb{F}_2$, where variables belong to $\mathbb{F}_2$. Therefore multiple of monomials in this ring is defined as follow:

$$m_1 \otimes m_2 = x_1^{i_1} \cdots x_n^{i_n} \otimes x_1^{j_1} \cdots x_n^{j_n} = x_1^{\max(i_1, j_1)} \cdots x_n^{\max(i_n, j_n)}.$$

Introduction

Problem: find all the common zeros in F

$$f_1(x_1, \ldots, x_n) = 0$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$f_m(x_1, \ldots, x_n) = 0$$

Among the connected questions of practical interest there are:

- Is the system consistent?
- If yes, how many solutions are there?
- How to eliminate some subset of variables?
- What are the solutions?

Algorithmic methods

1. Gröbner bases (Buchberger ' 1965)
2. Involutive bases (Gerdt, Blinkov ' 1998)

Gröbner Bases application

The general strategy of the Gröbner bases approach is as follows. Given a set F of polynomials that describes the problem:

$$F = \{f_1, f_2, \ldots, f_m\}, \quad f_i \in R(R')$$

we transform F into another set G of polynomials with certain more appropriate for further research properties called a Gröbner basis, such that F and G are "equivalent" and G is "simpler" than F.

As a result:

- Because of some special properties of Gröbner basis, many problems that are difficult for general F are "easier" for Gröbner basis.
- There is an algorithm for transforming an arbitrary F into G.
- The solution of the problem for G can often be easily translated back into a solution of the problem for F.

Equivalence of G and F

Equivalence means generation of the same polynomial ideal by both sets (bases) of polynomials:

$$Ideal(F) = Ideal(G)$$

where

$$Ideal(F) := \{\sum_{i=1}^{i=m} h_i f_i \mid f_i \in F,\ h_i \in R(R')\}$$

$$Ideal(G) := \{\sum_{i=1}^{i=m} h_i g_i \mid g_i \in G,\ h_i \in R(R')\}$$

It follows, in particular, that different bases of a given ideal have the same set of roots (variety).

Gröbner bases can and were applied successfully to:

- coding theory
- cryptography
- partial differential equations
- symbolic summation and integration
- statistics
- computer geometry and graphics
- computer aided design
- numerics (e.g. wavelets and difference schemes generation)
- system theory (e.g. control theory)
- quantum computing
- .....

Contents

Buhberger's algorithm

Algorithm: Gröbner Basis(F)

Input: $F \in R(R') \setminus \{0\}$ – finite set of polynomials
Output: G – Gröbner bases of $\mathrm{Id}(F)$

1: $B := \{[i,j] : 1 \leq i < j \leq \mathrm{length}(G)\}$
2: **while** $B \neq \emptyset$ **do**
3: $[i,j] :=$SelectPair(B, G)
4: $B := B \setminus \{[i,j]\}$
5: $h :=$NormalForm(Spoly$(G_i, G_j), G)$
6: **if** $h \neq 0$ **then**
7: $G := G \cup \{h\}$;
8: $B := B \cup \{[i, \mathrm{length}(G)] : 1 \leq i < \mathrm{length}(G)\}$
9: **end if**
10: **end while**
11: **return** G

Example

G	B
$\{x^3 - 2xy, x^2y + x - 2y^2\}$	$\{(1,2)\}$
$\{x^3 - 2xy, x^2y + x - 2y^2, -x^2\}$	$\{(1,3),(2,3)\}$
$\{x^3 - 2xy, x^2y + x - 2y^2, -x^2, -2xy\}$	$\{(2,3), (1,4),(2,4),(3,4)\}$
$\{x^3 - 2xy, x^2y + x - 2y^2, -x^2, -2xy, x - 2y^2\}$	$\{(1,4),(2,4),(3,4), (1,5),(2,5),(3,5),(4,5)\}$

10 reductions, 7 unnecessary reductions

Contents

Involutive algorithm

Algorithm: JanetBasis(F)

choose $f \in F$ such that $\mathrm{lm}(f) = \min\{\mathrm{lm}(F)\}$
$G := \{f\}$
$Q := F \setminus G$
while $Q \neq \emptyset$ **do**
    choose $p \in Q$ such that $\mathrm{lm}(p) = \min\{\mathrm{lm}(Q)\}$
    $Q := Q \setminus \{p\}$
    $h :=$JanetNormalForm(p, G)
    if $h \neq 0$ **then**
        $G := G \cup \{h\}$
        for all $q \in G$ and $x \in NM_J(q, G)$ **do**
            $Q := Q \cup \{q \cdot x\}$
        end for
    end if
end while
return G

Example

G	Q
$\{x^2y + x - 2y^2\}$	$\{x^3 - 2xy\}$
$\{x^2y + x - 2y^2, x^3 - 2xy\}$	$\{x^3y + x^2 - 2xy^2\}$
$\{x^2y + x - 2y^2, x^3 - 2xy, x^2\}$	$\{x^3, x^2y\}$
$\{x^2y + x - 2y^2, x^3 - 2xy, x^2, 2xy\}$	$\{x^2y, 2x^2y\}$
$\{x^2y + x - 2y^2, x^3 - 2xy, x^2, 2xy, -x + 2y^2\}$	$\{2x^2y, -x^2 + 2xy^2, -xy + 2y^3\}$
$\{x^2y + x - 2y^2, x^3 - 2xy, x^2, 2xy, -x + 2y^2\}$	$\{-x^2 + 2xy^2, -xy + 2y^3\}$
$\{x^2y + x - 2y^2, x^3 - 2xy, x^2, 2xy, -x + 2y^2\}$	$\{-xy + 2y^3\}$
$\{x^2y + x - 2y^2, x^3 - 2xy, x^2, 2xy, -x + 2y^2, 2y^3\}$	$\{\}$

6 reductions, 2 unnecessary reductions

Contents

Ring $\tilde{R}$

Every monomial order in $\tilde{R}$ is not an admissible one:

$$m_1 = x_1^{i_1} \cdots x_n^{i_n},\ m_2 = x_1^{j_1} \cdots x_n^{j_n},\ m_1 \neq m_2 \Longrightarrow m_1 \prec m_2 \text{ or } m_1 \succ m_2$$

$$m_3 = \operatorname{lcm}(m_1, m_2) = x_1^{\max(i_1, j_1)} \cdots x_n^{\max(i_n, j_n)}$$

$$m_1 \cdot m_3 = x_1^{\max(i_1, \max(i_1, j_1))} \cdots x_n^{\max(i_n, \max(i_n, j_n))} = m_3$$

$$m_2 \cdot m_3 = x_1^{\max(j_1, \max(i_1, j_1))} \cdots x_n^{\max(j_n, \max(i_n, j_n))} = m_3$$

A single polynomial basis in $\tilde{R}$ is not a Gröbner basis for sure.
Example: $\langle xy + x + 1 \rangle = \langle x + 1, y \rangle$.

Contents

Ideals in $\tilde{R}$ and R'

Theorem

Let us have $\{f_1, \ldots f_m\}$ – a finite set of polynomials in $\tilde{R}$. Using homomorphism from R' to $\tilde{R}$ a basis $\{\tilde{f}_1, \ldots, \tilde{f}_m, B\}$, where $\tilde{f}_i$ is the original of f_i in R' and $B = \{x_1^2 + x_1, \ldots, x_n^2 + x_n\}$, can be obtained. Applying any algorithm gives us a Gröbner basis – $\{\tilde{g}_1, \ldots, \tilde{g}_k, B'\}$, $B' \subseteq B$. Then the set $\{g_1, \ldots, g_k\}$ (g_i – image of $\tilde{g}_i$) is the required Gröbner basis of ideal $\langle f_1, \ldots, f_m \rangle$ in the ring $\tilde{R}$.

Therefore:
from now on we use the ring R' instead of $\tilde{R}$ for constracting desired Gröbner bases.

Contents

Vectorization

1-bit vectorization:

$$x_0 x_1 x_4 x_5 = \underbrace{\langle 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \rangle}_{64(128)\,bits}$$

2-bit vectorization:

$$x_0 x_1^2 x_4 x_5 = \underbrace{\langle 00\ 00\ 00\ 01\ 01\ 00\ 10\ 01 \rangle}_{64(128)\,bits}$$

Benchmarks

Used software and hardware

CoCoA 4.6, Singular 3.0.2, Mathematica 5.0,
GBF2 0.2.3 Buhberger and GBF2 0.3.5 Involutive.
Machine: 2xOpteron-242 (1.6 Ghz) with 6Gb RAM under Gentoo Linux 2005.1, gcc-4.1.0 compiler.

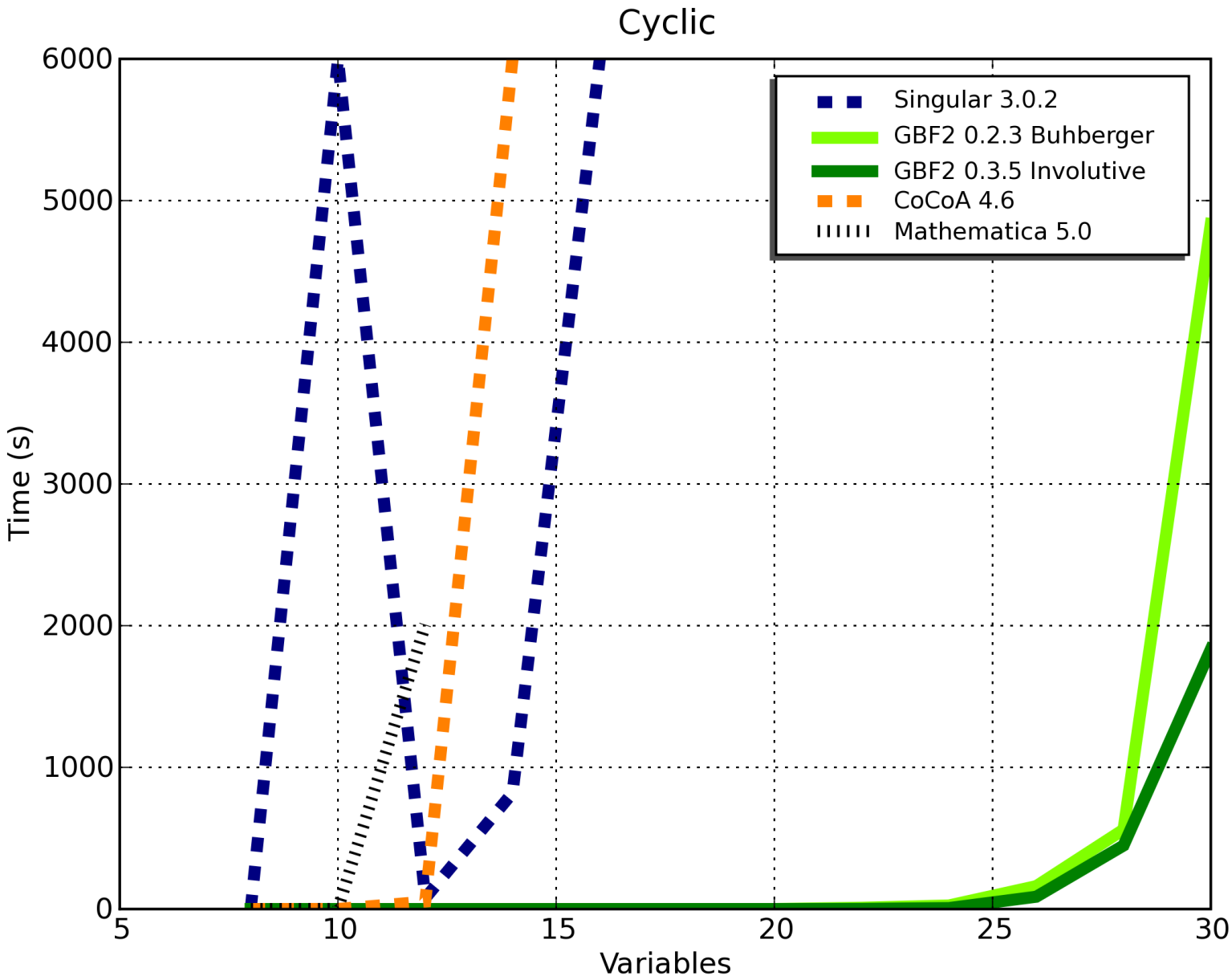
Used series

Cyclic, Eco, Redcyc, Redeco, Noon, Katsura – stantard series (http://www-sop.inria.fr/saga/POL).
Series Life: a single polynomial of variables $x_0, \ldots, x_i$

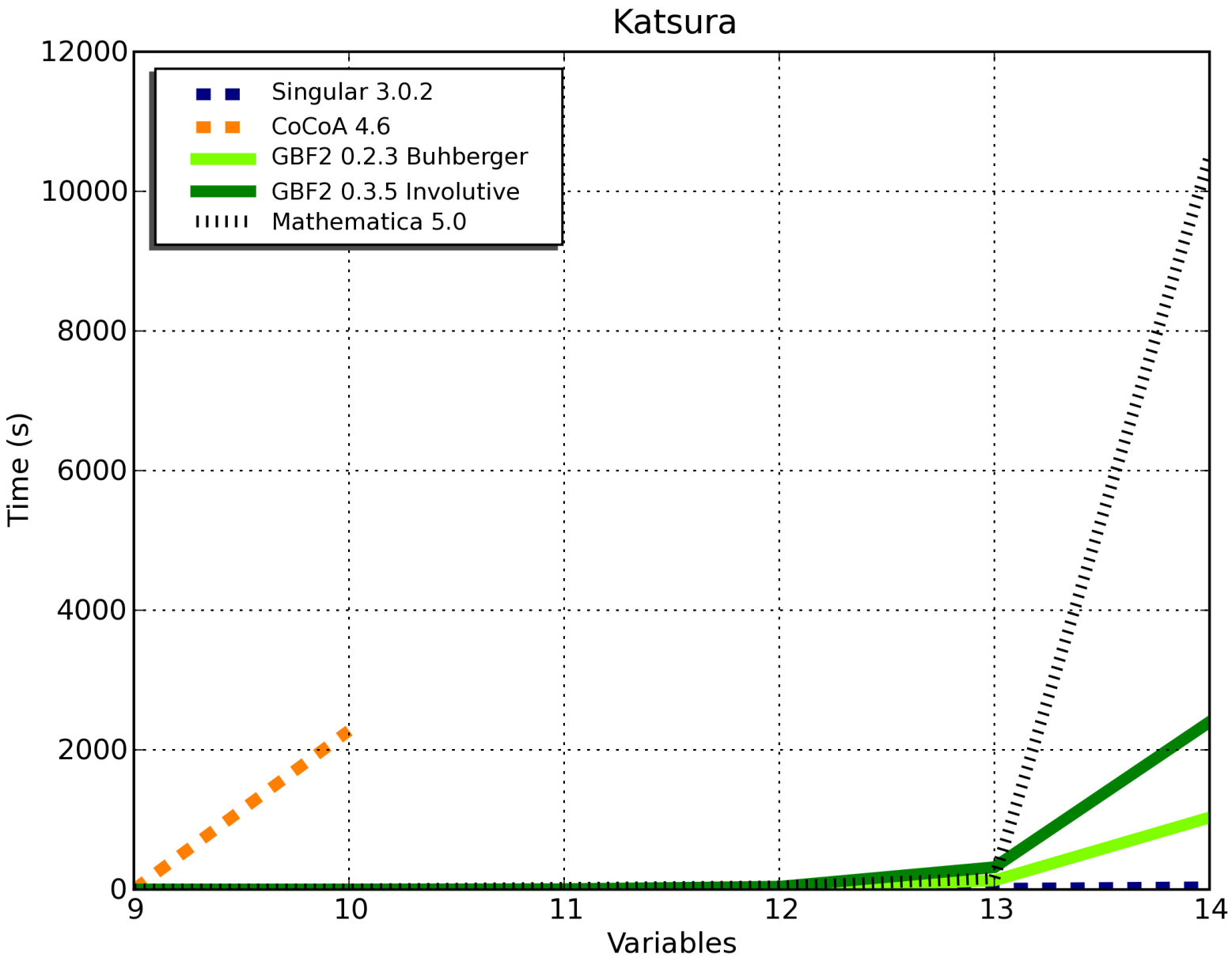
$$x_i + x_{i-1}(\sigma_{i-2} + \sigma_{i-3} + \sigma_3 + \sigma_2) + \sigma_{i-2} + \sigma_3$$

where σ_k is the $k-$th symmetric polynomial of variables $x_0, \ldots, x_{i-2}$.

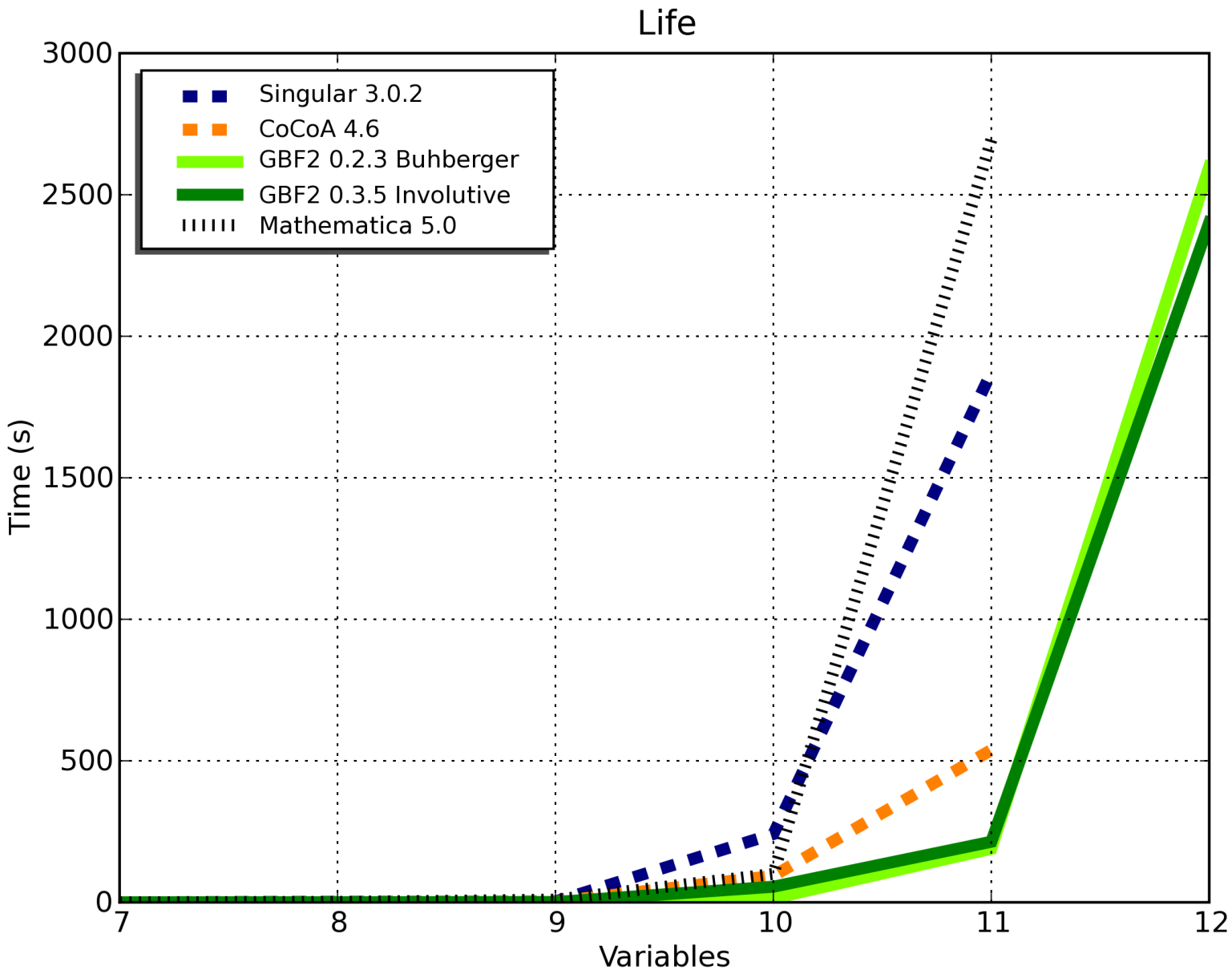
Cyclic series



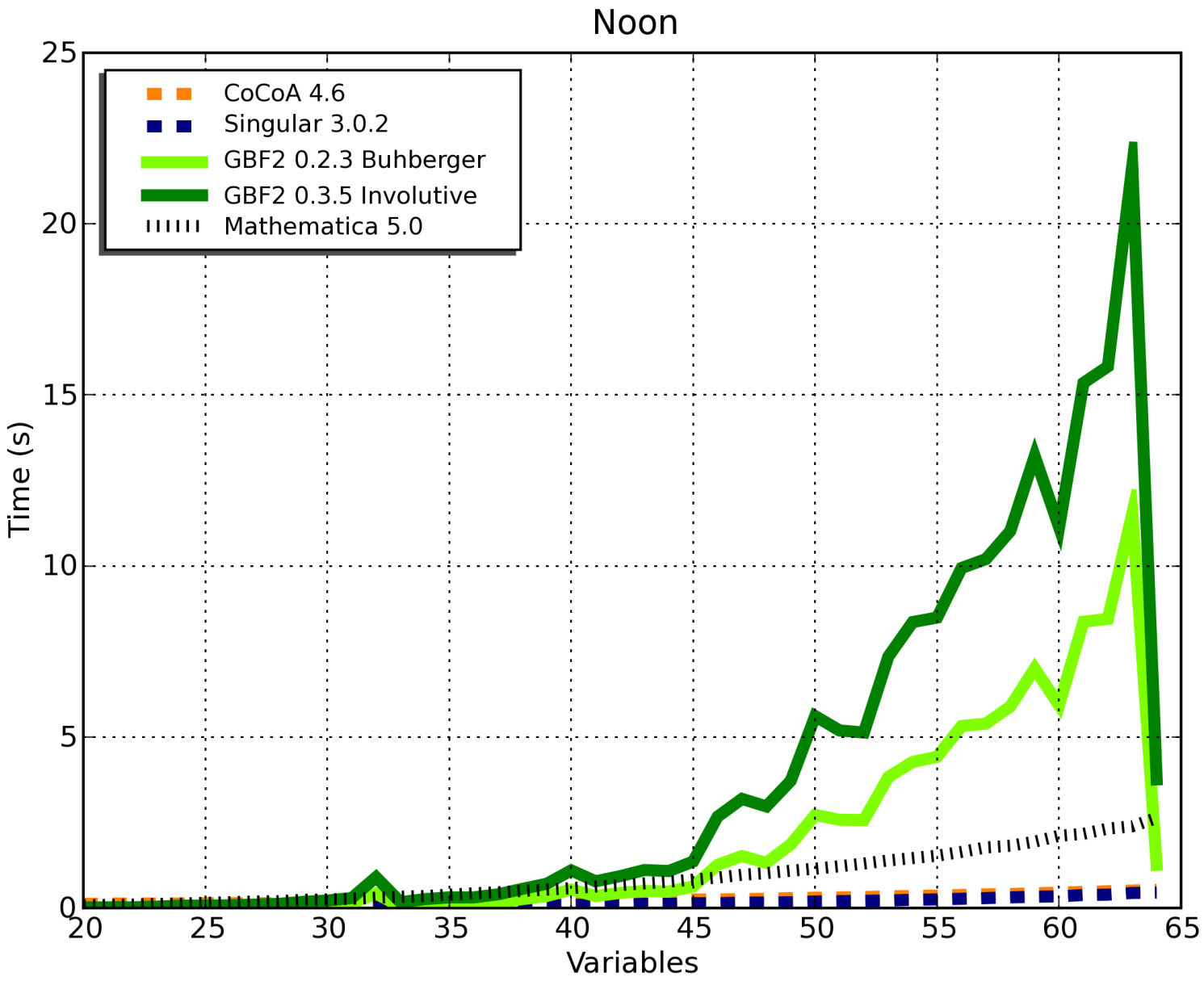
Katsura series



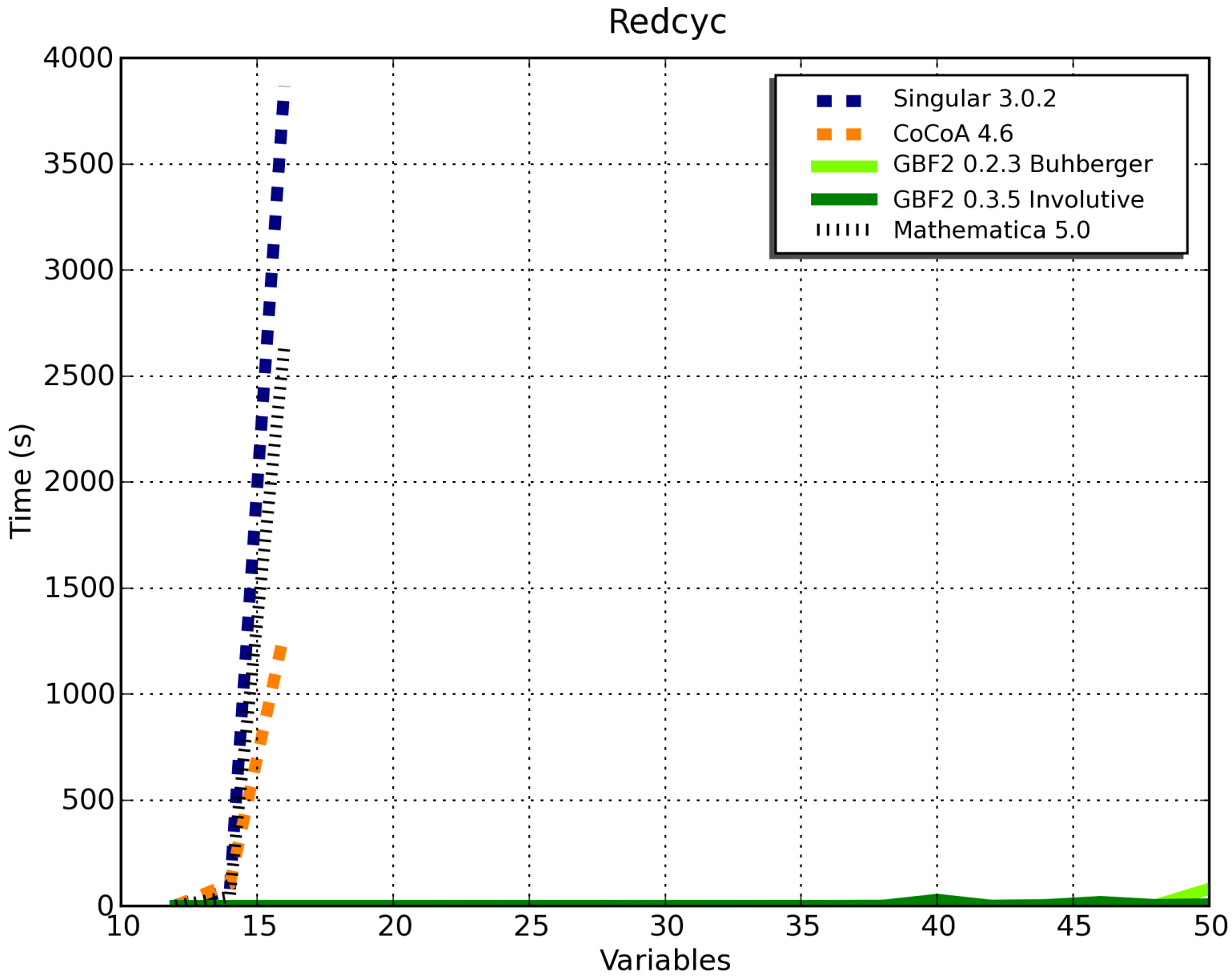
Life series


Life
Singular 3.0.2
CoCoA 4.6
GBF2 0.2.3 Buhberger
GBF2 0.3.5 Involutive
Mathematica 5.0
Time (s)
Variables
0
500
1000
1500
2000
2500
3000
7
8
9
10
11
12

Noon series


Noon
CoCoA 4.6
Singular 3.0.2
GBF2 0.2.3 Buhberger
GBF2 0.3.5 Involutive
Mathematica 5.0
Time (s)
Variables
0
5
10
15
20
25
20
25
30
35
40
45
50
55
60
65

Redcyc series



Further developments

- Improvement of inner data structures (1-bit vectorization, etc.)
- Search for more appropriate strategy for selection of
  1. critical pairs (Buchberger's algorithm)
  2. nonmultiplicative prolongations (Involutive algorithm)
- Implementation of the FGLM algorithm for converting of degree-reverse-lexicographical Gröbner or involutive bases (that are computed much faster then their pure lexicographical counterparts) into the corresponding lexicographical bases.
- Using division other that Janet one (for example Pommaret, etc.) and more appropriate for computation over $\mathbb{F}_2$
- Installation as a program module in the specialized computer algebra system GINV (see http://invo.jinr.ru).