Shell Exploration & Production

Algebraic Computations on Noisy, Measured Data

Daniel Heldt, Sebastian Pokutta, Hennie Poulisse

Based on Joint On-Going Work:
Daniel Heldt, Martin Kreuzer, Sebastian Pokutta, Hennie Poulisse

Contents

- Industrial Application of Computer Algebra
- ApVI Calculated

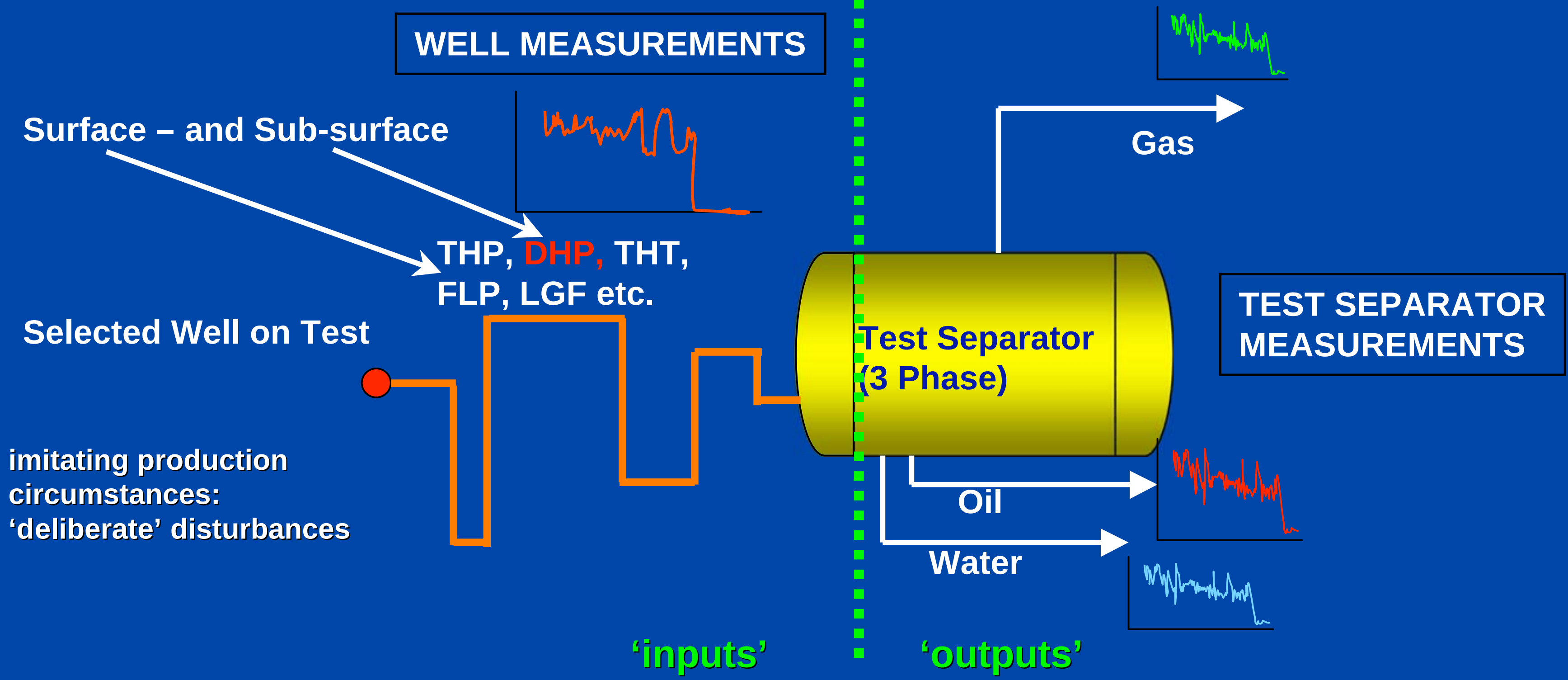
Shell Exploration & Production

The ‘Champions Field’, South-Chinese Sea, offshore Brunei


transportation tubing
well heads
headers
(test- and bulk) separators

Shell Exploration & Production

Production Relation for an Oil Well



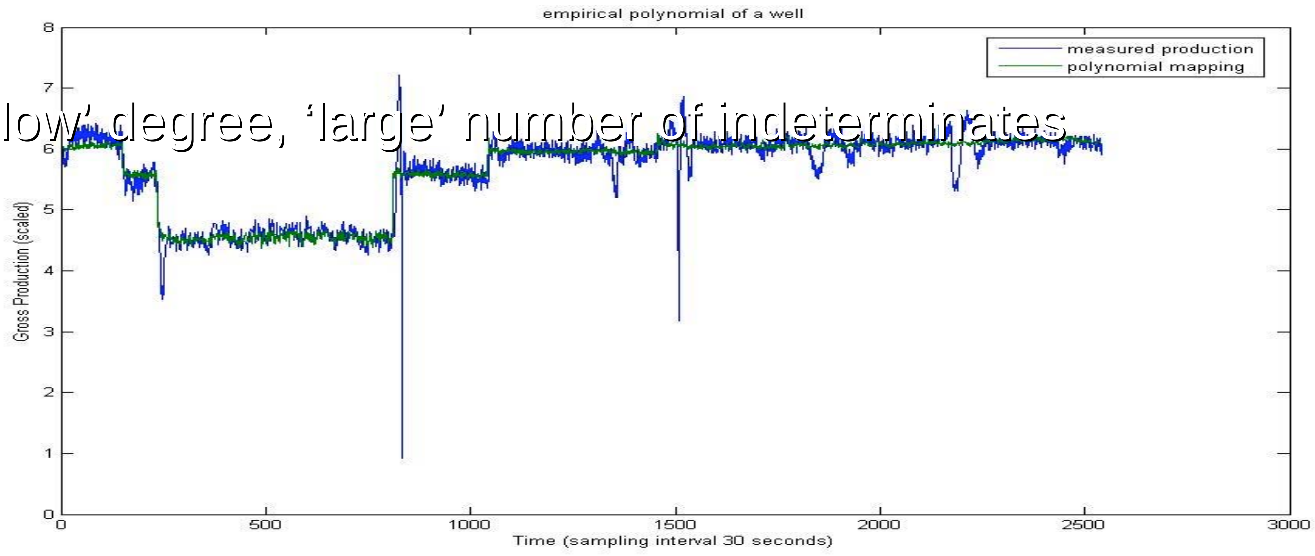
output = function of inputs

output = polynomial function

variables of polynomial function { **input** / **physical relation between inputs** }

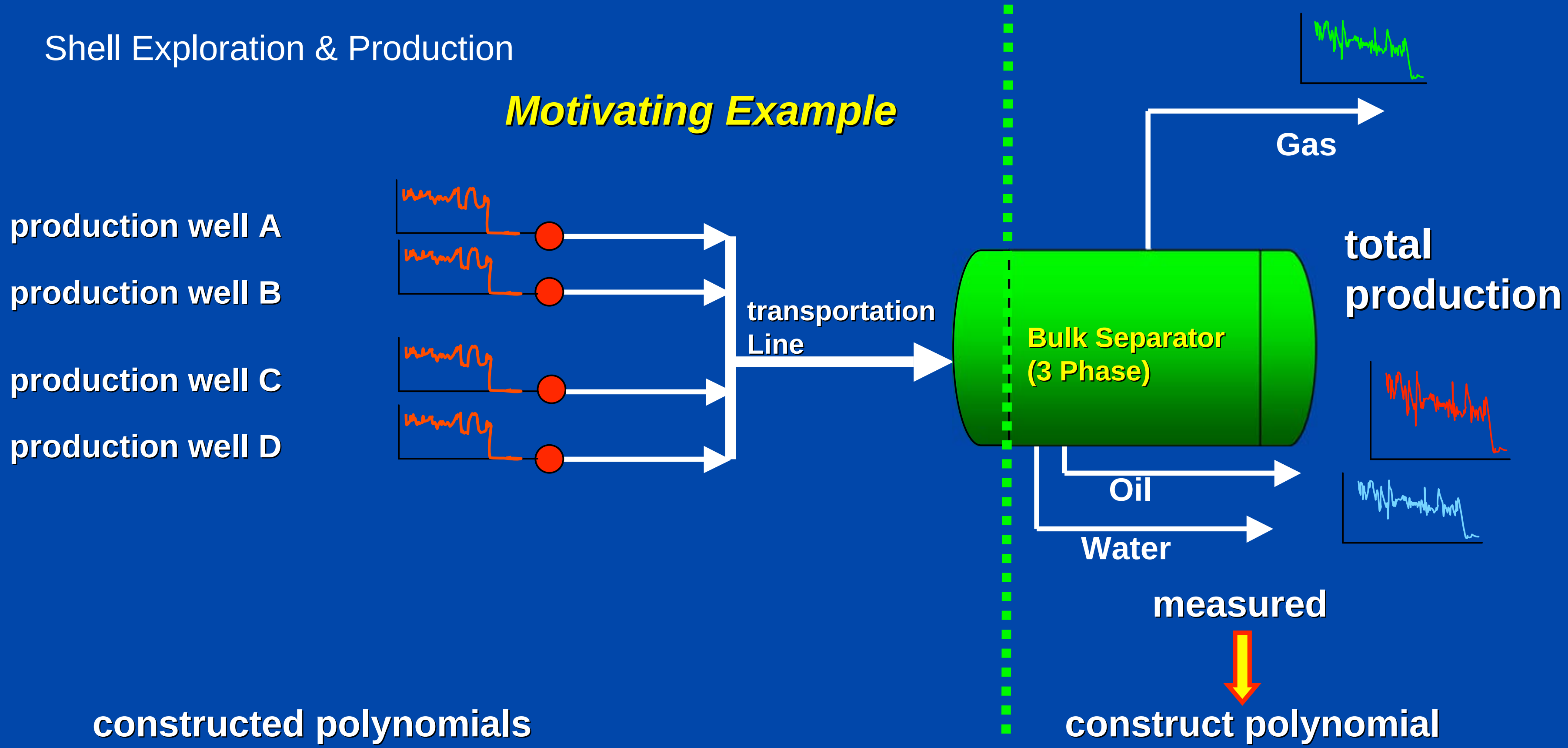
example of a polynomial well production

$$
\begin{aligned}
production = {} & 2.9232 * (\sqrt{}((DHP_1 - DHP_{tu}) * DHP_{tu}) * FLP) - \\
& 2.7075 * (\sqrt{}((DHP_1 - DHP_{tu}) * DHP_{tu}) * DHP_1) + \\
& 2.9517 * \sqrt{}((DHP_{tu} - THP) * THP) - \\
& 1.2158 * (\sqrt{}((DHP_{tu} - THP) * THP) * THP) + \\
& 0.3856 * (\sqrt{}((THP - FLP) * FLP) * DHP_{tu}) + \\
& 2.5594 * (\sqrt{}((DHP_1 - DHP_{tu}) * DHP_{tu}) * DHP_{tu})
\end{aligned}
$$



'low' degree, 'large' number of indeterminates

Motivating Example



$$Total\ Production \in Ideal\ Generated\ by\ Separate\ Productions$$

$$prod_{total} = g_A prod_{well\ A} + \cdots + g_D prod_{well\ D}$$

polynomials: interactions between productions

construction of well production polynomial

$$P = \mathbb{R}[x_1, \ldots, x_n]$$

indeterminates

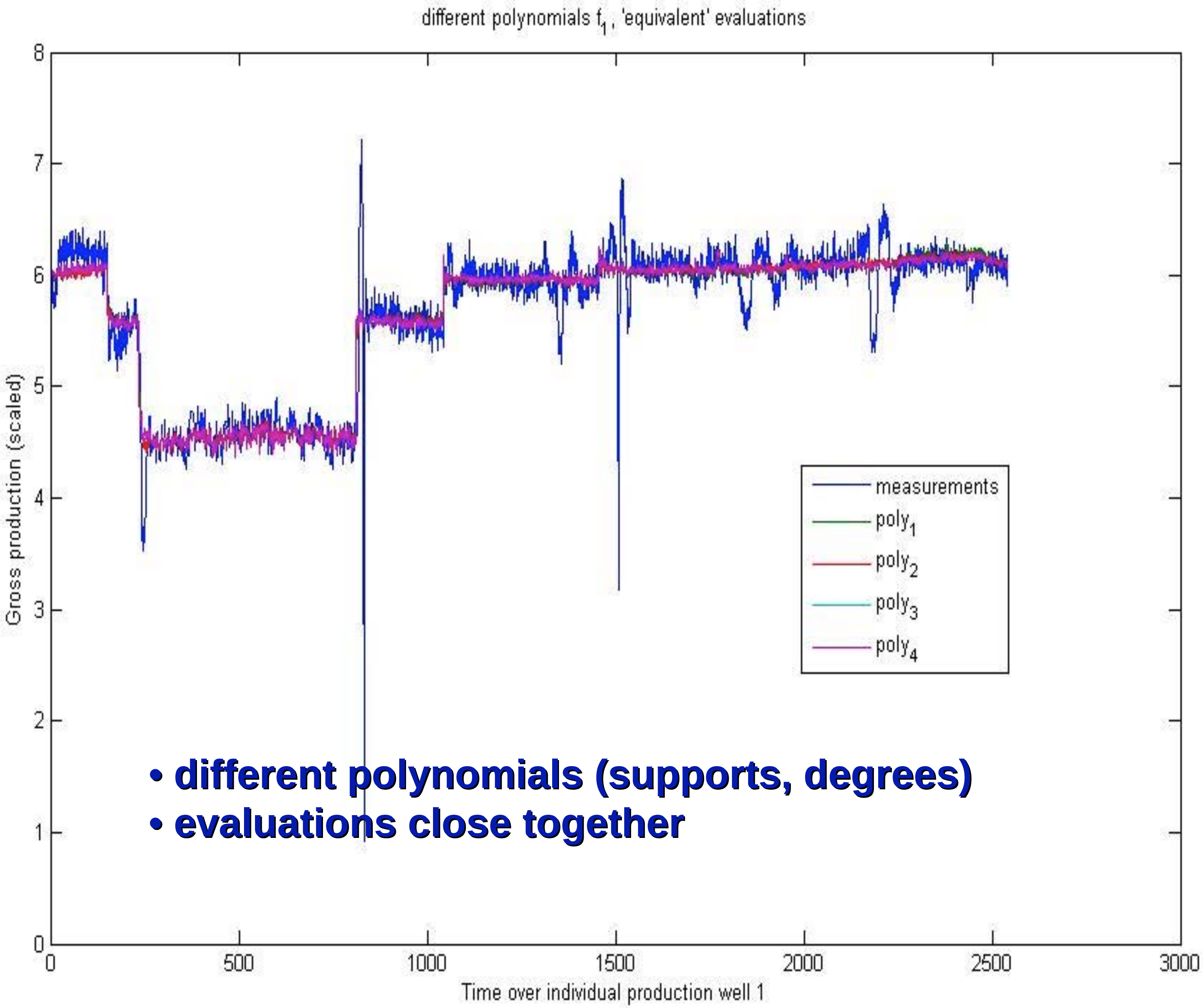
$$x_1 = THP,\ x_2 = \sqrt{((DPT - P_{tubing})P_{tubing})}, \ldots$$

production polynomial *f* has to be fitted to a set of points:

$$\mathbb{X} = \{p_1, \ldots, p_s\} \subset \mathbb{R}^n$$

evaluations of indeterminates at first data point

number of points in the order of thousands not uncommon



- **different polynomials (supports, degrees)**
- **evaluations close together**

relations in the data, among the indeterminates

$$
\begin{aligned}
f &\in P \; polynomial\ estimated\ from\ noisy\ data \\
\psi_{p_i} &: P \rightarrow \mathbb{R} \; evaluation\ homomorphism:\ h \mapsto h(p_i) \\
\delta &\in \mathbb{R}^+ \\
E_f &= \{g \in P \mid |g(p) - f(p)| \leq \delta \forall p \in \mathbb{X}
\end{aligned}
$$

~~$constructed\ polynomials\ 'per\ set'\ E_f$??~~

transitivity fails

Small Polynomials:

$$r \in P \; small \; polynomial \;\; \Leftrightarrow |\psi_p(r)| \leq \delta \, \forall \, p \in \mathbb{X}$$

Vanishing Ideal

$$I(\mathbb{X}) := \{g \in P \mid \psi_p(g) = 0 \, \forall \, p \in \mathbb{X}\}$$

example

$$p_1 = \cdots = p_5 \in \mathbb{R}^2$$
$$f(p_1) = \cdots = f(p_5) = 0$$

→ **univariate polynomial**

perturb points

→ **polynomial of degree 5 vanishing on points**

- **remove the purturbations first before constructing the polynomials**
- **construct polynomial allowing it to pass ‘close by’ prescribed points**

delta-Approximate Vanishing Ideal

$$I_\delta(\mathbb{X}) \quad : \quad \exists G \; with \; normalized \; coefficient \; vectors$$
$$s.t. \quad |g(p)| \leq \delta \, \forall p \in \mathbb{X} \; and \; g \in G$$

of course we still have:

$$\exists g \in I_\delta(\mathbb{X}) \, , \, |g(p)| > \delta \, , \; for \; some \; p \in \mathbb{X}$$

Best thing one can do:

Empirical Ring:

$$Q := P/I_\delta(\mathbb{X})$$

Empirical Polynomial:

$$f \in Q$$

Shell Exploration & Production

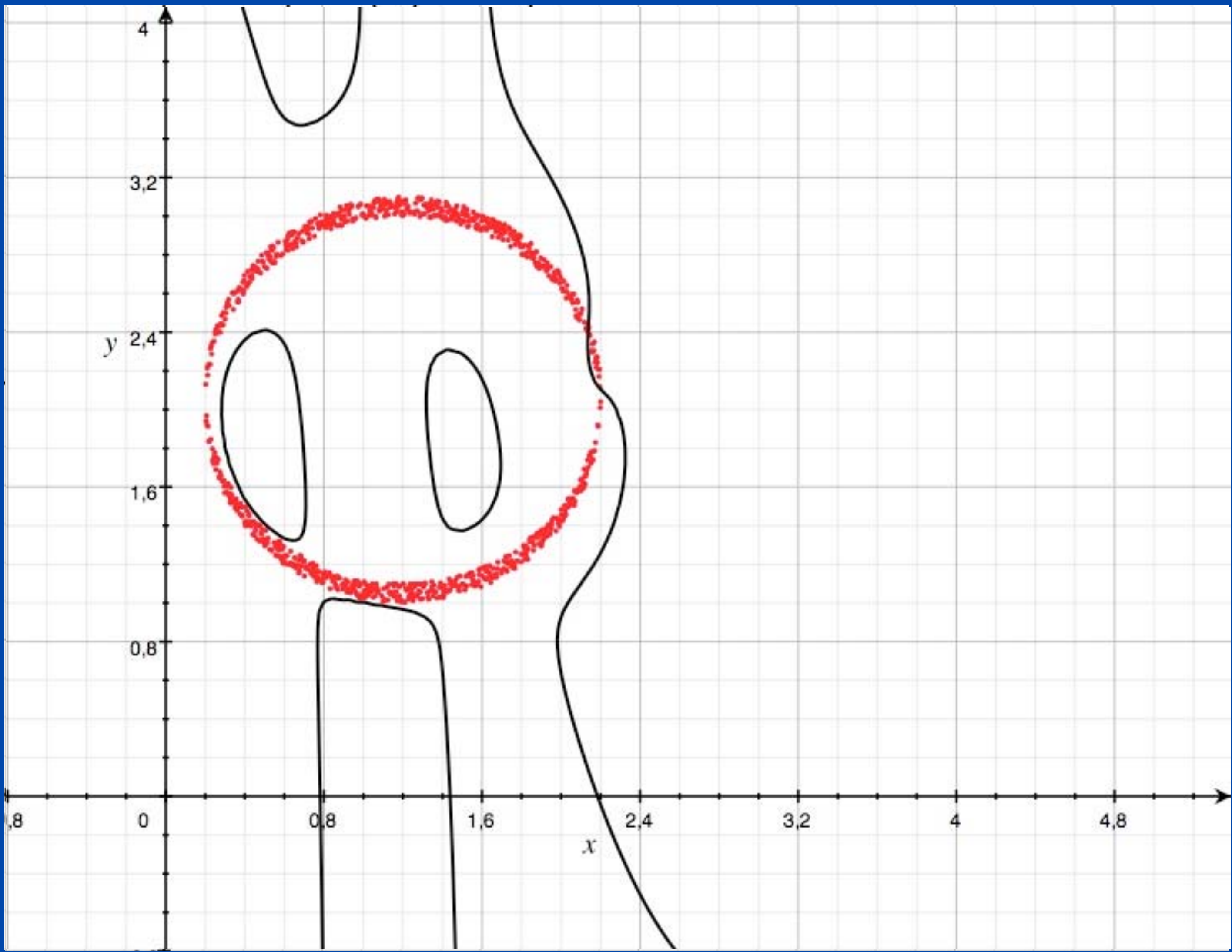
HOW TO CALCULATE THE

ApVI

The Buchberger–Möller algorithm:

1. Set $\mathcal{O} = \emptyset, Eval(\mathcal{O}) = \emptyset$ and $G = \emptyset$.
2. If $\mathbb{T}^n \setminus \mathrm{Lt}_\sigma(G) = \emptyset$, stop; otherwise let t be the set's minimal element.
3. Compute $v = (t(p_1), \ldots, t(p_m))^T$ via evaluating t on the point set $\mathbb{X}$.
4. If $v \in Eval(\mathcal{O})$, find a representation of v and add the corresponding polynomial to G. Then go to Step 2.
5. Add t to $\mathcal{O}, v$ to $Eval(\mathcal{O})$ and go to step 2.

Classical Version



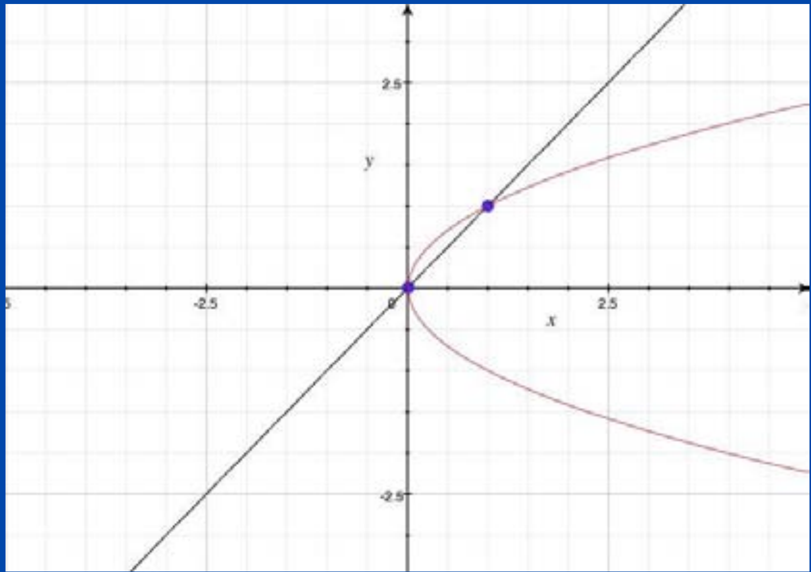
Drawbacks:

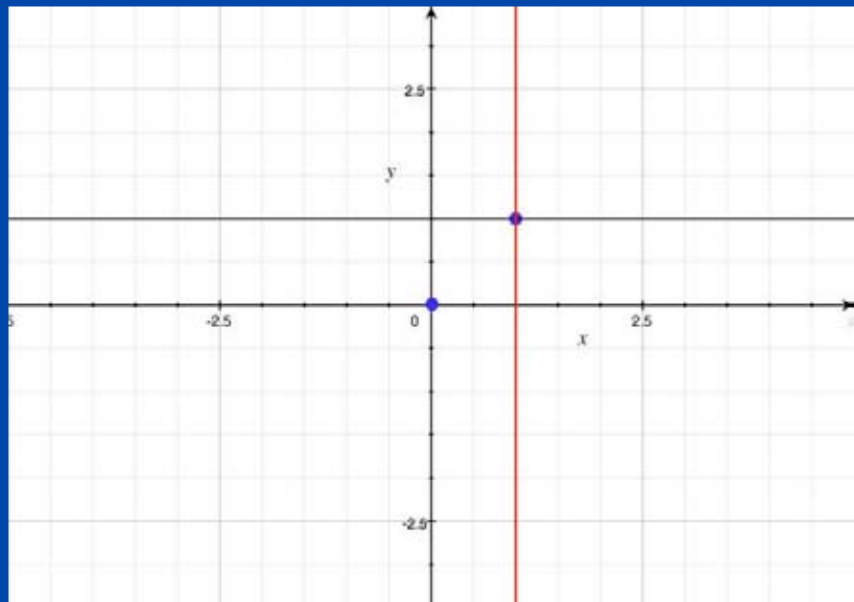
- The found relations are far away from vanishing on the points:
  - numerical error prevent accurate calculation
  - exact solution would result in an O with #O = #P (1000 here!!)
  - The exact relations (for the perturbed data) are usually of very high degree

How to overcome these problems:

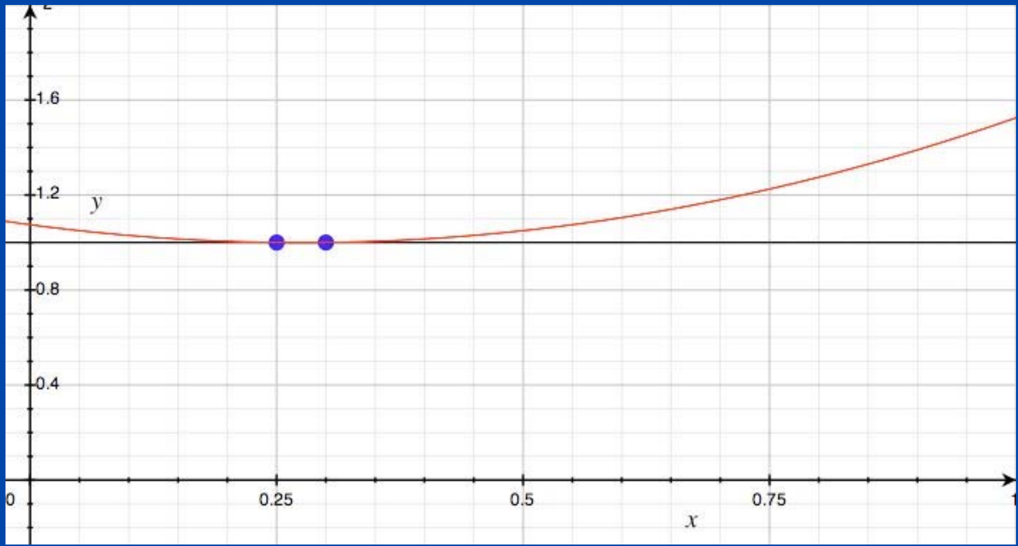
- Do not force the relations to pass through every point, but demand passing „close by“
- Allow points which lie “close together” to be “melt together”
- “Divide the good ones from the bad ones”
- Process blocks rather than single elements to (hopefully) prevent sub-optimal solutions and speed-up computations

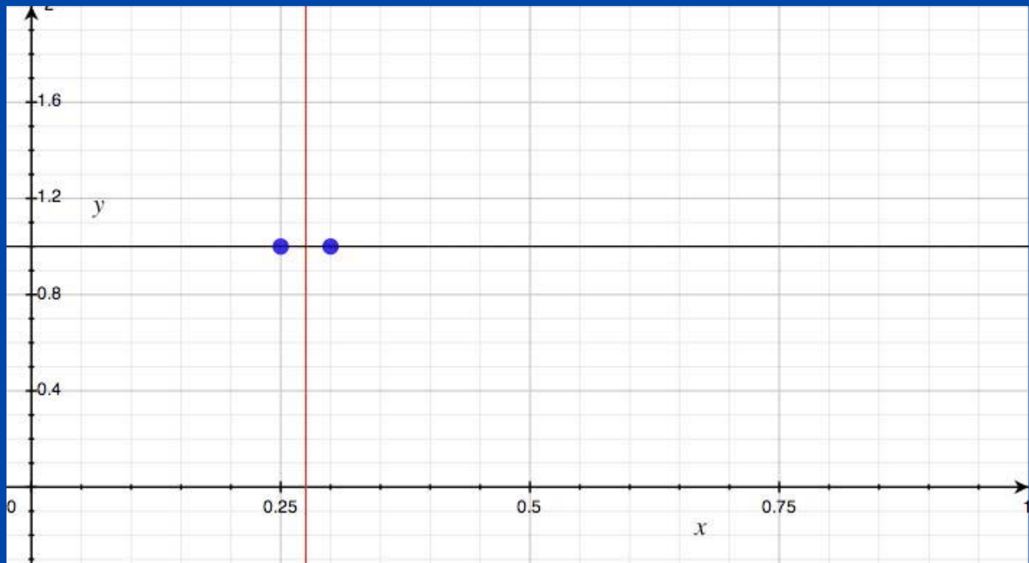
Shell Exploration & Production





Shell Exploration & Production

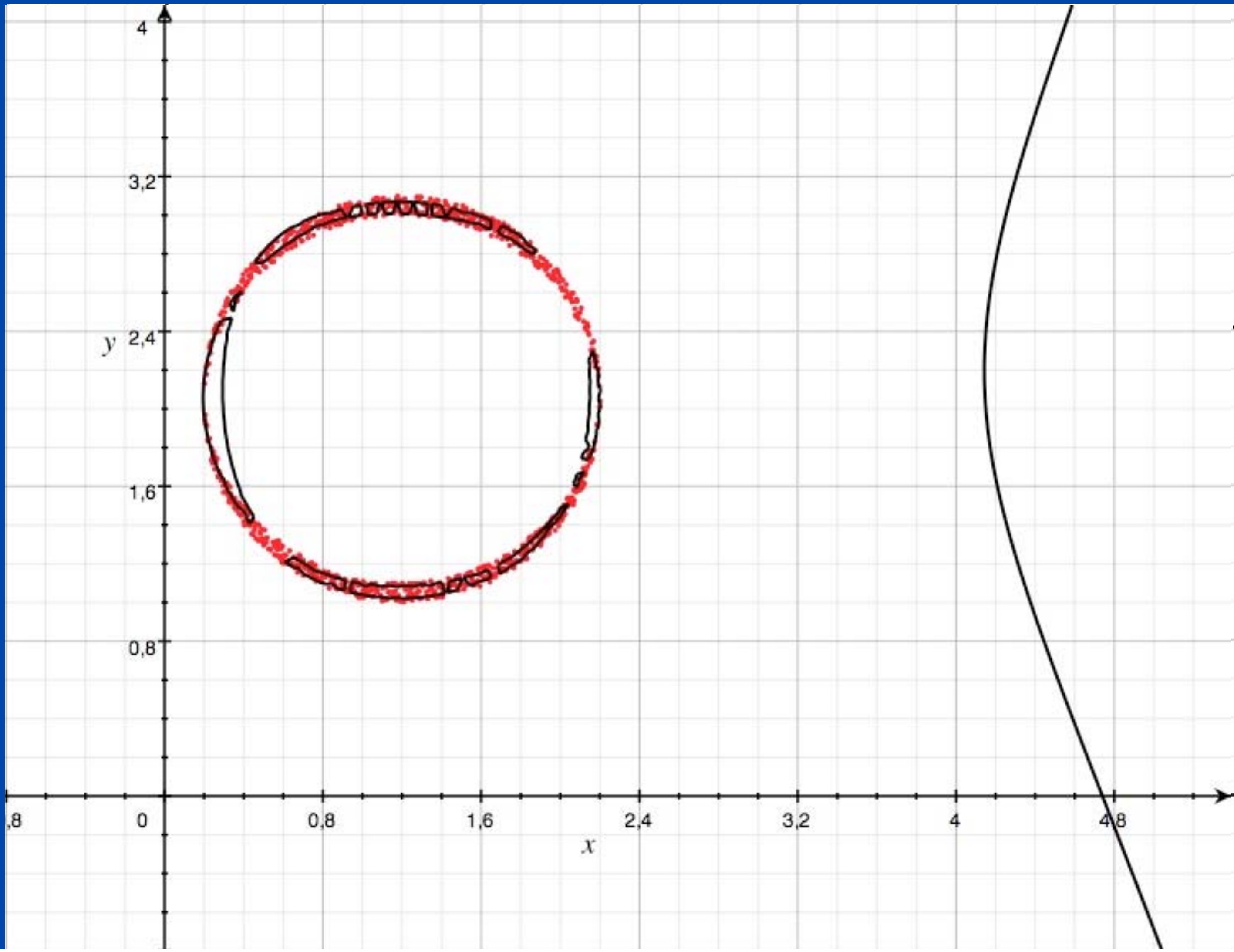




An approximate Buchberger–Möller algorithm:

1. Set $\mathcal{O} = \emptyset, Eval(\mathcal{O}) = \emptyset, G = \emptyset$ and $d = 0$.
2. If $\mathbb{T}^n \setminus \mathrm{LT}_\sigma(G) = \emptyset$, stop; otherwise let $T = \{t_1, \ldots, t_s\}$ contain the set's polynomials of degree d.
3. Compute the matrix $V = (v(t_1), \ldots, v(t_n))$, where $v(t_i) = (t_i(p_1), \ldots, t_i(p_m))^T$ is t_i's evaluation on the point set $\mathbb{X}$.
4. Compute the singular values of the matrix $(V \;\; Eval(\mathcal{O}))$ and a basis of all singular vectors with singular values $< \varepsilon$. Represent these basis as a set of (reduced) polynomials and add them to G.
5. Add all terms in $T \setminus \mathrm{LT}_\sigma(G)$ to $\mathcal{O}$ and their evaluations to $Eval(\mathcal{O})$, increment d and go to step 2.

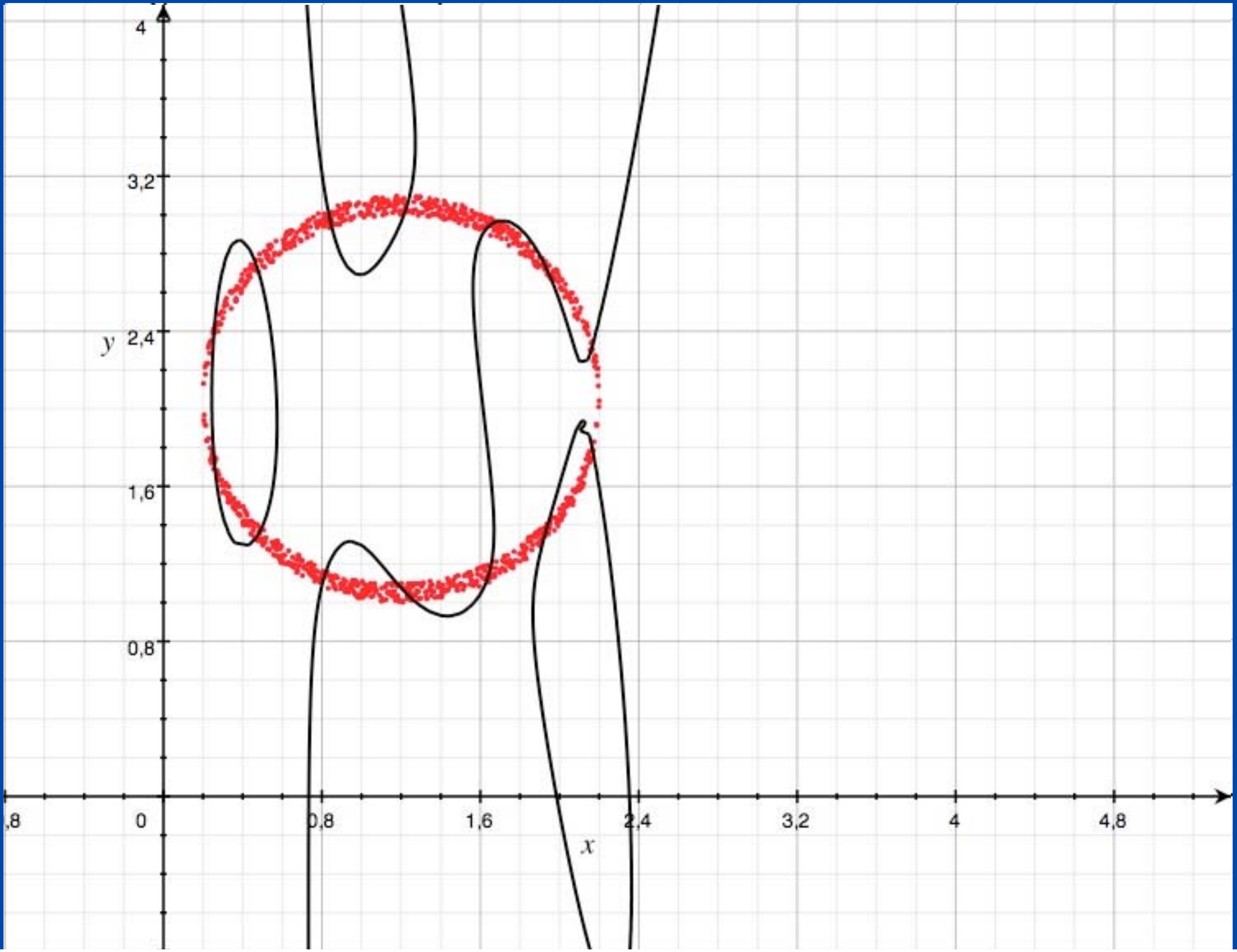
Approximate Vanishing Ideal with Singular Value Decomposition



Truncate below eps = 0.001

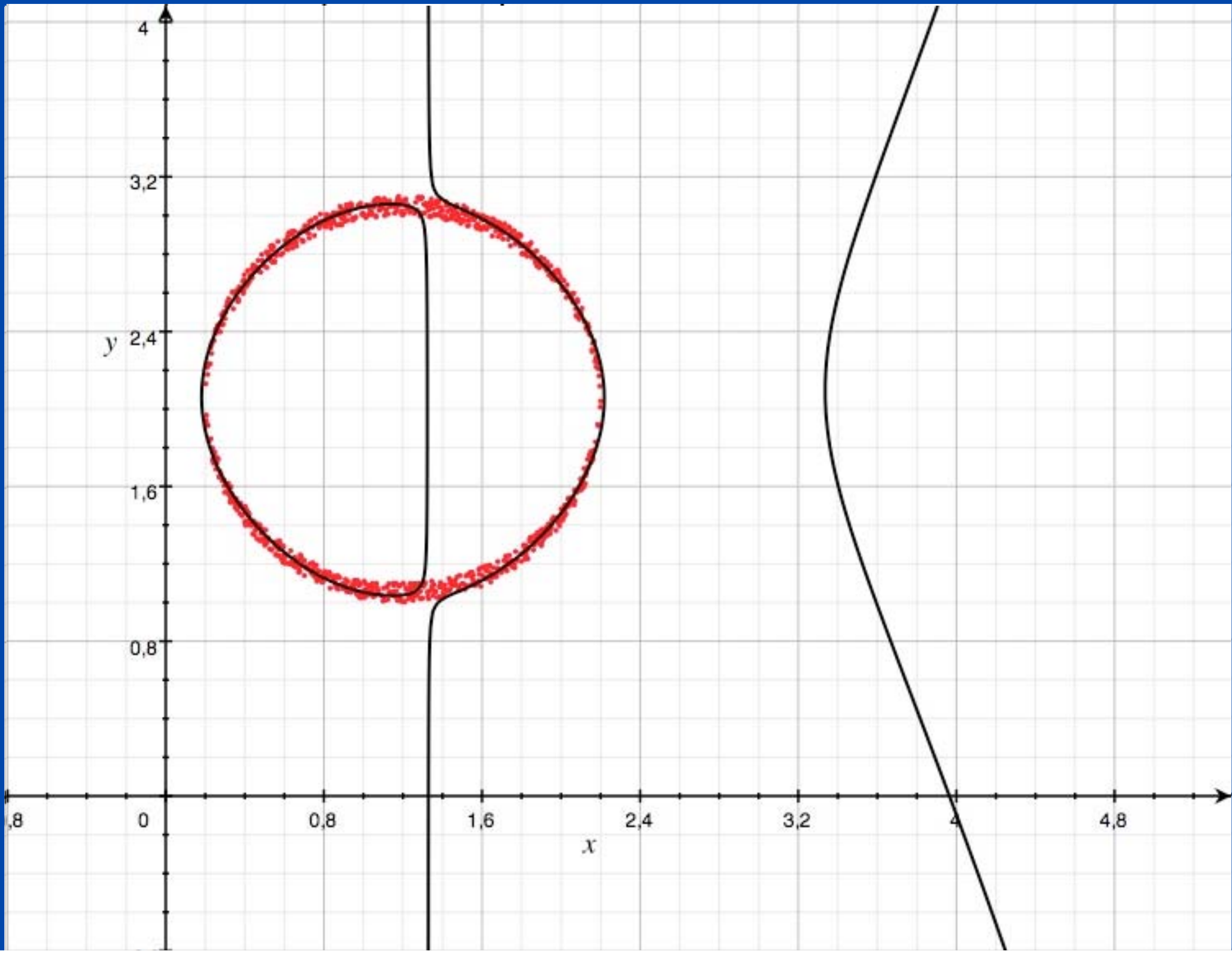
Shell Exploration & Production

Approximate Vanishing Ideal with Singular Value Decomposition



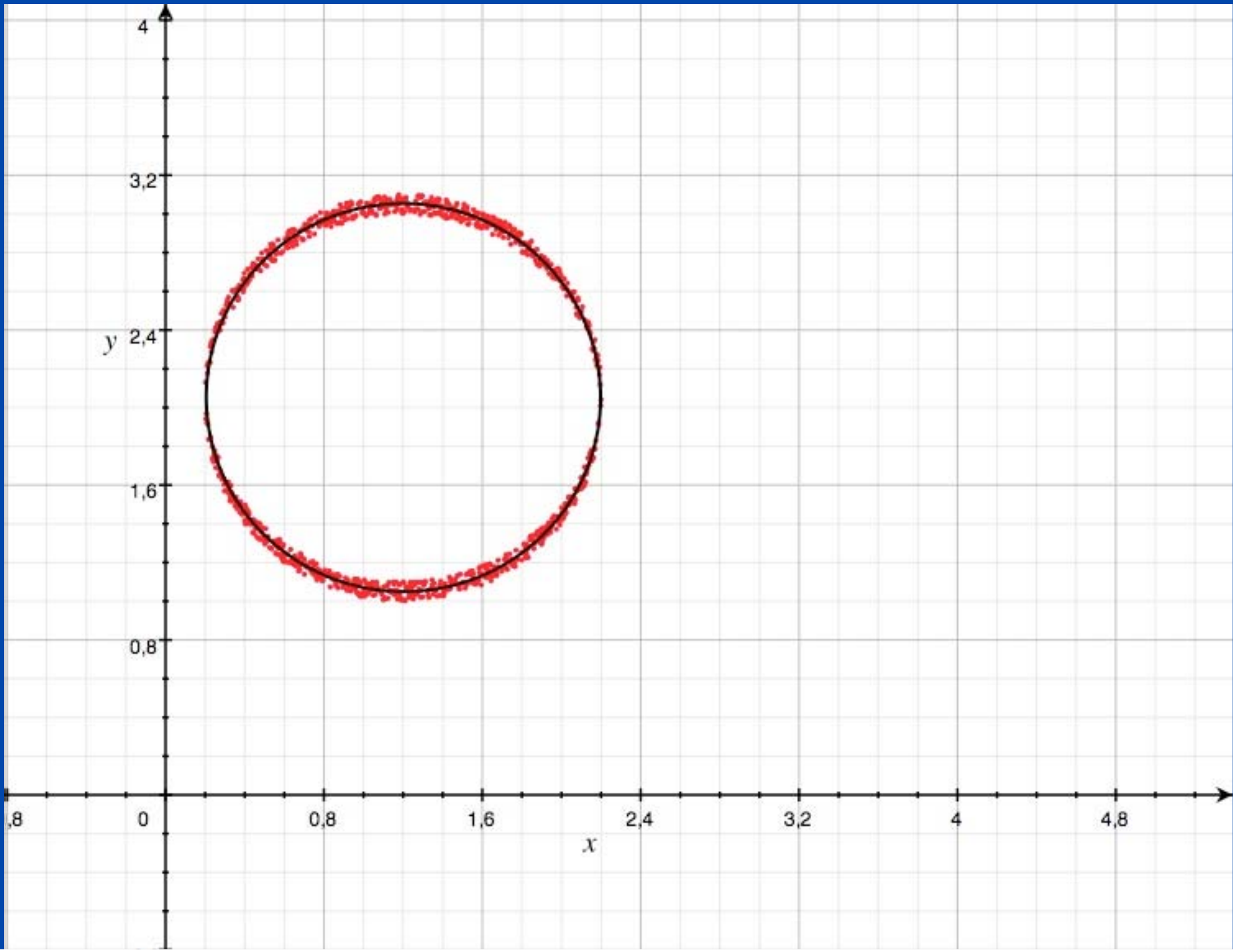
Truncate below
eps = 0.01

Approximate Vanishing Ideal with Singular Value Decomposition



Truncate below eps = 0.1

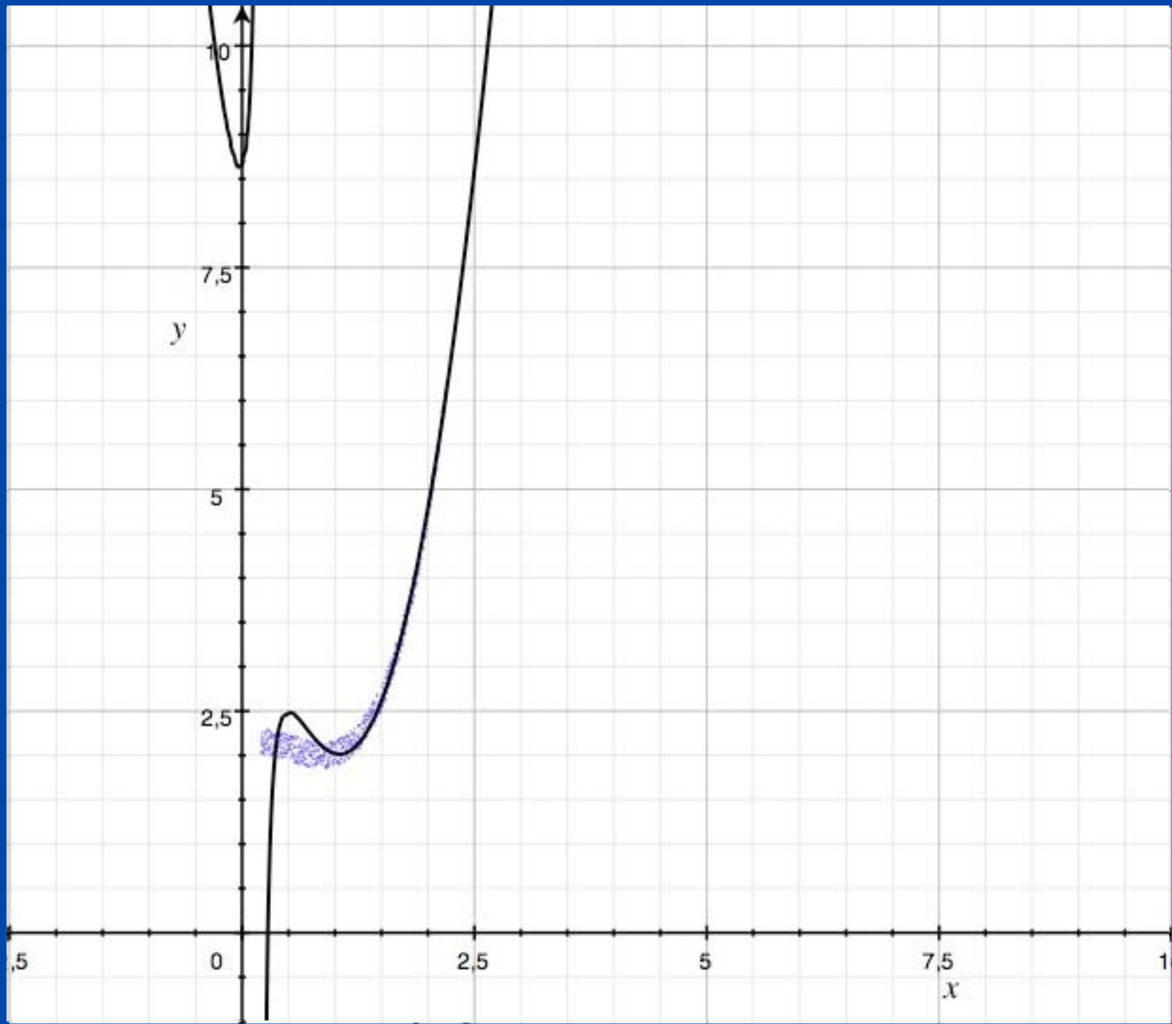
Approximate Vanishing Ideal with Singular Value Decomposition



Truncate below eps = 0.5

Another (short) example...

Approximate Vanishing Ideal with Singular Value Decomposition



Classical Version

Truncate below eps = 0.7

An approximate Buchberger–Möller algorithm for border basis:

1. Set $\mathcal{O}s = \emptyset, Eval(\mathcal{O}) = \emptyset, G = \emptyset$ and $d = 0$.
2. If $(\mathcal{O} \cdot \{x_1, \ldots, x_n\})_d$ is empy, stop; otherwise let $T = \{t_1, \ldots, t_s\}$ contain the set's polynomials of degree d.
3. Compute the matrix $V = (v(t_1), \ldots, v(t_n))$, where $v(t_i) = (t_i(p_1), \ldots, t_i(p_m))^T$ is t_i's evaluation on the point set $\mathbb{X}$.
4. Compute the singular values of the matrix $(V\,Eval(\mathcal{O}))$ and a basis of all singular vectors with singular values $< \varepsilon$. Represent these basis as a set of (reduced) polynomials and add them to G.
5. Add all terms in $T \setminus \mathrm{LT}_\sigma(G)$ to $\mathcal{O}$ and their evaluations to $Eval(\mathcal{O})$, increment d and go to step 2.

Timings (GB-Version) (2024 points, 9 indets)

	ϵ	# in basis	max deg	max m-error	mean m-error	calc. time
1	1	15	3	0.015	0.0025	0.77 s
2	0.5	19	3	0.0063	$8.1262 \cdot 10^{-4}$	0.84 s
3	0.1	24	3	$6.9260 \cdot 10^{-4}$	$2.7851 \cdot 10^{-4}$	0.82 s
4	0.01	39	4	$9.0761 \cdot 10^{-5}$	$1.7142 \cdot 10^{-5}$	1.43 s
5	0.001	59	4	$1.5547 \cdot 10^{-6}$	$1.0135 \cdot 10^{-6}$	2.34 s
6	0.0001	82	4	$4.8458 \cdot 10^{-7}$	$1.0858 \cdot 10^{-7}$	4.04 s
7	0.00001	125	4	$8.4620 \cdot 10^{-8}$	$8.0948 \cdot 10^{-9}$	7.31 s
8	0.0 (full)	260	5	$4.6954 \cdot 10^{-11}$	$2.0193 \cdot 10^{-11}$	22.04 s

TABLE 1. Results (real-world data set) for calculating a DegLex Gröbner basis

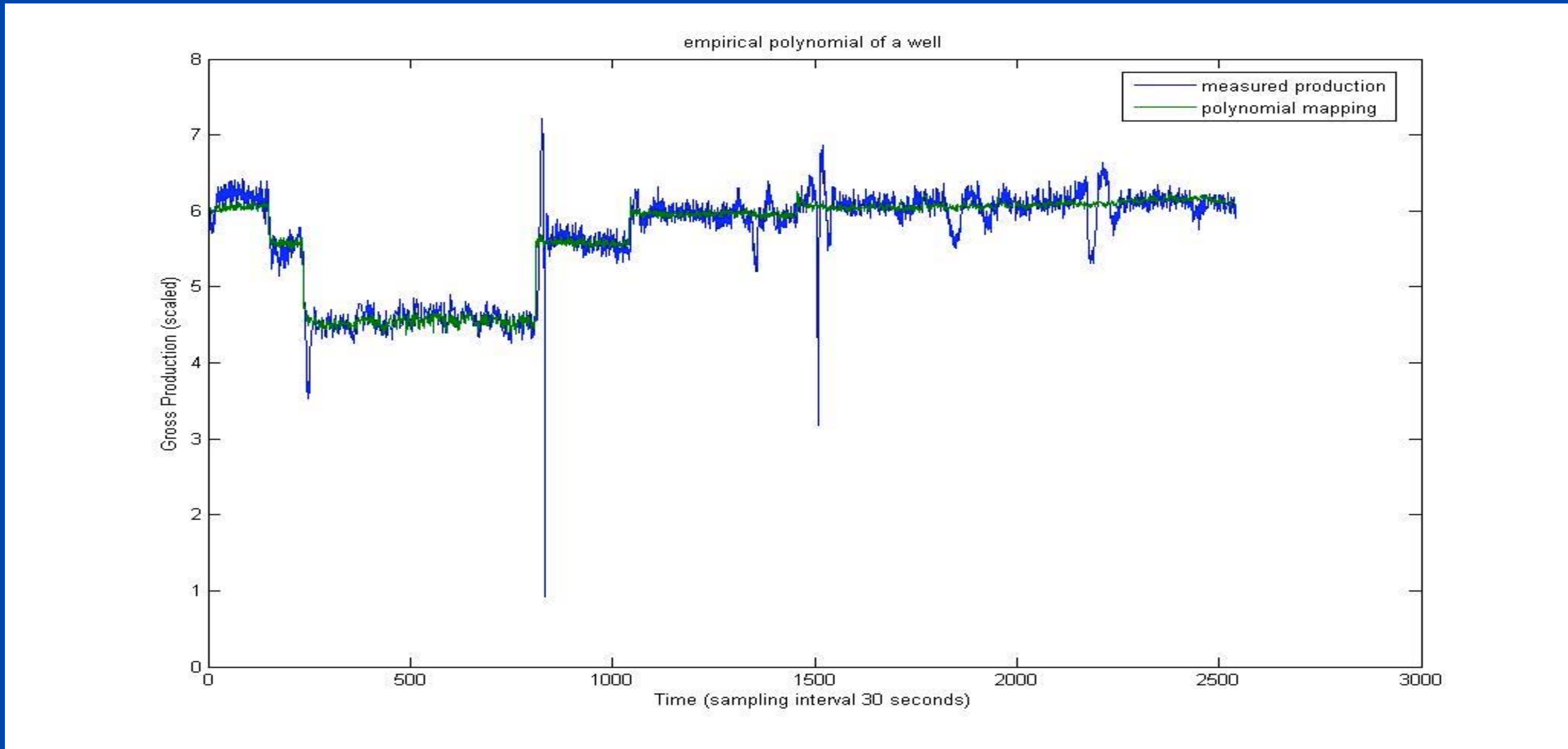
Timings (BB-Version)
(2024 points, 9 indets)

	ϵ	# in basis	max deg	max m-error	mean m-error	calc. time
1	1	94	6	0.0115	$5.5523 \cdot 10^{-4}$	1.86 s
2	0.5	88	5	0.0063	$3.4916 \cdot 10^{-4}$	1.72 s
3	0.1	185	8	$6.9626 \cdot 10^{-4}$	$3.1908 \cdot 10^{-5}$	4.19 s
4	0.01	163	5	$9.0761 \cdot 10^{-5}$	$3.3018 \cdot 10^{-6}$	4.60 s
5	0.001	223	5	$1.5547 \cdot 10^{-6}$	$2.1958 \cdot 10^{-7}$	8.07 s
6	0.0001	306	6	$4.8458 \cdot 10^{-7}$	$2.4311 \cdot 10^{-8}$	13.55 s
7	0.00001	345	5	$8.4620 \cdot 10^{-8}$	$3.4753 \cdot 10^{-9}$	20.52 s
8	0.0 (full)	539	5	$3.6442 \cdot 10^{-11}$	$1.3308 \cdot 10^{-11}$	53.37 s

TABLE 2. Results (real-world data set) for calculating a border basis

Shell Exploration & Production

Application within Shell



Calculate relations for the productions...

Typical datasize: 2000-5000 points in up to 15 indets

Application within steel industries

Application fields:

- predict production quality
- automated quality assurance

Typical datasize: 10000-15000 points in 10-15 indets

- *solution time < 238.73 s*
- *#G = 464, #O = 391*

Acknowledgement:

C. Fassino and J. Abbott worked in parallel on an algorithm which also computes almost vanishing ideals, but with a different approach... See:

C. Fassino. An Approximation to the Gröbner Basis of Ideals of Perturbed Point. Preprint (2006)

Shell Exploration & Production

Thank you! !